DESIGN OF CFD CODE USING HIGH LEVEL PROGRAMMING PARADIGMS: FREE SURFACE FLOWS WITH ARBITRARILY MOVING RIGID BODIES

OLEG I. GUBANOV

Design of CFD code using high level programming paradigms: Free surface flows with arbitrarily moving rigid bodies

by

Oleg I. Gubanov

A thesis submitted in partial fulfillment
of the requirements for the degree of
Masters of Science (Computational Science)

School of Graduate Studies
Memorial University of Newfoundland
St. John's, Newfoundland and Labrador

Spring 2007

Your file Votre référence
ISBN: 978-0-494-31249-0
Our file Notre référence
ISBN: 978-0-494-31249-0

Abstract

The objective of this thesis is to design and develop a numerical simulation tool for the class of unsteady free surface flows with arbitrarily moving rigid bodies. The major technical issue encountered in the solution of free surface problems is the appearance of numerical instabilities that arise due to the description of the mesh movement to track the moving boundaries, nonlinearity of the governing equations and boundary conditions implementation at the free surface and rigid body surface.

In the present computational model, a fluid flow governed by the full Navier-Stokes equations is only modeled within a two-dimensional computational domain. The motion of the air is neglected and the effect of the ambient pressure exerted on the fluid by the air is taken into consideration. The method of solution is based on a finite volume discretization of the unsteady Navier-Stokes equations in their pressure-velocity formulation on a fixed Cartesian grid. A special integral form of the governing equations is derived by extending the Reynolds transport theorem and then applying it to control volumes containing a fluid interface for the first time. The combined volume of fluid and fractional area/volume obstacle representation method, and the cut cell method are employed to track the fluid-air and fluid-body interfaces. This combined method is based on techniques used in the commercial fluid dynamics simulation package `FLOW-3D` developed by Flow Science, Inc. A sparse linear system in pressure and velocity components is solved by using the generalized minimal residual method with ILUT preconditioner to advance the simulation of unsteady flow in time.

Creative use of object oriented programming, data abstraction and template metaprogramming paradigms enables us to implement principle computational fluid dynamics concepts, directly, into the computer code developed in this thesis. This results in a computer program that is easier to understand and maintain. The `C++` programming language is chosen as the implementation language since it provides direct support of these modern programming paradigms. Automatic generation of the program code from templates augmented with compile time optimizations and function inlining reduces the size of a manually written code up to four times and meets the high level code efficiency requirements at the `Fortran` level.

This numerical simulation tool is applied to the problem of unsteady, laminar, two-dimensional flow of a viscous incompressible fluid past a transversely oscillating circular cylinder in the presence of a free surface. The results of this thesis represent the first comprehensive numerical study conducted on this problem to date. The code validations are presented in special cases and good comparisons with previous experimental and numerical results are obtained.

Acknowledgments

I would like to express my sincere appreciation to my supervisor, Dr. Serpil Kocabiyik for her distinguished supervision, invaluable guidance, constant support and close collaboration throughout the course of my M.Sc. study at Memorial. Her supervisory style paved the way for becoming an independent researcher and provided continual enthusiasm. Throughout the thesis-writing period, Dr. Kocabiyik's expert advice and proof-reading have helped me to analyze the results of this research and to reduce the number of stylistic, grammatical and typographical mistakes in this thesis.

I especially thank my colleagues and friends, Larisa Mironova and Vitaly Khoriakov, for their constant support and invaluable discussions of the research issues and technical details.

I acknowledge the support of the Department of Mathematics and Statistics for providing the computational facilities which were essential to this work.

I would also like to thank Dr. George Miminis and Dr. Sharene Bungay for reviewing my thesis.

Financial assistance in the form of a graduate fellowship offered by Memorial University of Newfoundland is also gratefully acknowledged. Part of this financial support was provided by a Natural Sciences and Engineering Research Council of Canada (NSERC) research grant received by Dr. Serpil Kocabiyik.

Contents

List of Tables

List of Figures

Nomenclature

$\mathbb{A}$	fractional area open to flow
A	amplitude of the cylinder oscillations ($= A^*/d$)
$a(t)$	angular displacement of the cylinder
C_L	lift coefficient ($= 2L^*/\rho d U^2$)
C_D	drag coefficient ($= 2D^*/\rho d U^2$)
$\overline{C}_L$	mean lift coefficient
$\overline{C}_D$	mean drag coefficient
d	cylinder diameter
D^*	drag force per unit length
f	forcing frequency of the cylinder oscillation ($= df^*/U$)
f_0	natural vortex shedding frequency ($= df_0^*/U$)
Fr	Froude number
F	volume of fluid function
$\underline{F}$	force coefficient ($= (C_D, C_L)$)
$\underline{F}^*$	force per unit length ($= (D^*, L^*)$)
$\underline{g}_0$	gravitational acceleration in inertial frame of reference, $\mathbb{X}$ ($= (0, g_0, 0)$)
$\underline{g}$	gravitational acceleration in grid frame of reference, $\mathbb{X}_g$ ($= (-\ddot{x}_g, g_0 - \ddot{y}_g, 0)$)
h	cylinder submergence depth ($= h^*/d$)
$\mathbb{I}$	fluid interface
$\underline{\underline{I}}$	identity tensor
L^*	lift force per unit length
$\underline{n}$	outward normal
p	fluid pressure

p_0	ambient air pressure
R	Reynolds number ($= Ud/\nu$)
S_{cv}	boundary of the computational cell V_{cv}
t	time ($= t^*U/d$)
T	period of cylinder oscillation ($= 1/f$)
U	uniform flow velocity
$\underline{u}$	fluid velocity vector ($= (u, v, 0)$)
$\mathbb{V}$	fractional volume open to flow
V_{cv}	computational cell
$V(t)$	material volume
$\partial V(t)$	boundary of the material volume
$\mathbb{X}$	inertial frame of reference
$\mathbb{X}_g$	grid reference frame
$\mathbb{X}_{\text{c}}$	reference frame that moves and rotates with cylinder
$x(t)$	streamwise displacement of the cylinder with respect to $\mathbb{X}_g$
$x_g(t)$	streamwise displacement of the grid with respect to $\mathbb{X}$
$y(t)$	transverse displacement of the cylinder with respect to $\mathbb{X}_g$
$y_g(t)$	transverse displacement of the grid with respect to $\mathbb{X}$

Greek Symbols

ρ	fluid density
μ	dynamic fluid viscosity
ν	kinematic fluid viscosity ($= \mu/\rho$)
$\underline{\underline{\sigma}}$	stress tensor

Superscripts

*	dimensional quantity
$\dot{}$	time derivative ($= d/dt$)
$\ddot{}$	second time derivative ($= d^2/dt^2$)

Abbreviations

CFD	computational fluid dynamics
FAVOR	fractional area/volume obstacle representation
PSD	power spectral density
VOF	volume of fluid

Chapter 1

Preliminary discussion and mathematical model

1.1 Introduction

This thesis describes the design and development of a numerical simulation tool within an object oriented software framework for the solution of a class of free surface problems with arbitrarily moving rigid bodies. This numerical simulation tool is applied to the problem of unsteady, laminar, two dimensional flow of a viscous incompressible fluid past a transversely oscillating circular cylinder in the presence of a free surface. The unsteady hydrodynamic forces on a moving body and resulting free-surface deformations are of considerable importance in many engineering applications and theoretical interest. Offshore structures, towing and rapidly deployable cables, water

vehicles, and power generation equipment that captures energy contained in water mass due to tides are all representative examples of where this model can find practical applications. A large number of bluff structures found in engineering applications are horizontal cylindrical structures with circular cross-sections.

The problem of flow past a cylinder, placed in a fluid flow of an infinite extent, has been a subject of interest to engineers and scientists for a great many years. The practical significance of this problem has led to a large number of fundamental studies, many of which are discussed in the comprehensive reviews of King (1977), Bearman (1984), Griffin and Hall (1991), Williamson (1996), Rockwell (1998), Williamson and Govardhan (2004), Sarpkaya (2004) and in book chapters by Naudascher and Rockwell (1993), Sumer and Fredsøe (1997), Zdravkovich (1997), Anagnostopoulos (2002). Relatively few studies have been conducted on the problem of flow past a cylinder in the presence of a free surface. This problem has been principally investigated in the experimental studies by Sheridan *et al.* (1997), Zhu *et al.* (2000), Carberry *et al.* (2001), Cetiner and Rockwell (2001), Carberry (2002).

Free surface problems including bluff bodies have also received attention from computationalists. Ananthakrishnan (1991) employed a finite difference method based on boundary fitted coordinates to investigate surface waves generated by translating two-dimensional surface-piercing bodies with emphasis on the assessment of the nonlinear viscous effects at a Reynolds number $\mathrm{R}\,(= LU/\nu)$ of 10^3 and at the range of Froude numbers $\mathrm{Fr}\left(= U/\sqrt{Lg}\right)$ between 0.2 and 0.8. Here, L is the draft of the body defined as the width of the body, U is the translational body velocity, g is the acceleration due to gravity, and ν is the kinematic viscosity of the fluid. Fekken

(2004) studied transverse oscillations of a rectangular body partly submerged into a fluid using the cut-cell method originally developed by Verstappen and Veldman (2000) for Cartesian non-boundary fitted grids, in conjunction with the volume of fluid (VOF) method due to Hirt and Nichols (1981). Simulations were carried out at a Reynolds number $\mathrm{R}\,(=\sigma B^2/\nu)$ of 10^3 and at Froude numbers $\mathrm{Fr}\left(=\sigma\sqrt{B/g}\right)$ of 1.0, 1.5 and 2.0. The dimensionless amplitude, a/B, is taken to be 0.1 and 0.2; the dimensionless draft of the body, d/B, is 0.5. Here, B is the beam of the body defined as the distance from the waterline to the bottom of the body, σ is the frequency of the body oscillations, ν is the kinematic viscosity of the fluid and g is the acceleration due to gravity.

To date, the numerical treatment of the problem of uniform flow past a circular cylinder in the presence of a free surface has remained relatively unexplored. To the author's knowledge, only numerical studies by Reichl (2001), Reichl *et al.* (2005) and Yan (2000) discuss stationary and oscillating circular cylinders in the presence of a free surface. Reichl *et al.* (2005) numerically investigated two-dimensional flow past a stationary cylinder close to a free surface at a Reynolds number $\mathrm{R}(=dU/\nu)$ of 180, where d is the cylinder diameter, U is the uniform free stream velocity and ν is the kinematic viscosity of the fluid. The simulations are conducted at Froude numbers $\mathrm{Fr}\left(=U/\sqrt{dg}\right)$ between 0.00 and 0.70 and gap ratios $h\,(=h^*/d)$ between 0.1 and 5.0. Here, g is the acceleration due to gravity and h^* is the distance between the top of the cylinder and the position of the undisturbed free surface. The simulations are carried out using the computational fluid dynamics software package FLUENT based on the finite volume method to solve the two-phase flow model as described by Hirt and Nichols (1981). The quadratic upstream interpolation for convective kinematics

method of Leonard (1979) is employed for the spatial discretization, which is second-order accurate. The temporal discretization is only first-order accurate as the VOF method is employed. Reichl *et al.*'s work demonstrated that the problem shares many features with flow past a cylinder close to a no-slip wall at low Froude numbers where the surface experiences little or no deformation. At larger Froude numbers in excess of $0.3 - 0.4$, surface deformation becomes substantial which results in significant surface vorticity altering the development of Strouhal vortices from the top shear layer, affecting wake skewness and suppressing the absolute instability. Reichl *et al.*'s work also demonstrated that the flow becomes more two-dimensional at smaller submergence depths and shedding ceases at very small gap ratios ($\sim 0.1 - 0.2$).

Yan (2000) considered the uniform, two-dimensional flow past a circular cylinder submerged beneath the free surface of a viscous fluid of infinite depth to investigate the influence of imposed steady current on the resulting steady streaming. The cylinder itself performs a prescribed small-amplitude oscillatory motion with amplitude A^* and frequency $\omega^*(= 2\pi f^*)$. Yan employed both numerical and perturbation schemes to calculate the viscous flow and focused on relatively slow flow by choosing the magnitude of the steady current as $O(\varepsilon)$ where $\varepsilon = A^*/a \ll 1$ with a being the cylinder radius. The viscous flow simulations are conducted at Reynolds numbers, $\mathrm{R}_b = \dfrac{a^2\omega^*}{\nu}$, of $O(1)$. Boundary conditions at the free surface and at large distances from the cylinder are obtained from the inviscid flow solution. This is done by expanding the velocity potential in powers of ε and using the boundary element method described by Riley and Yan (1996). Yan's investigation is based on the second-order accurate Crank-Nicolson scheme to solve the full unsteady Navier-Stokes equations in their stream function - vorticity formulation. An important finding of Yan's work is

that when the magnitude of the steady current is small, the Reynolds stress produced by the oscillation becomes significant in the neighbourhood of the cylinder which results in the induced steady streaming of the classic four-cell structure for rectilinear cylinder oscillations unlike the case when the magnitude of the steady current is relatively large. However, the circulation around the cylinder due to the induced steady streaming is almost independent of the magnitude of the steady current when the cylinder performs uniform orbital motion.

The major technical issue encountered in the solution of free surface problems is the appearance of numerical instabilities that arise due to the description of the mesh movement to track the moving boundaries, nonlinearity of the governing equations, and boundary conditions implementation at the free surface and rigid body surface. In the present computational model, fluid flow governed by the full Navier-Stokes equations is only modeled within a two-dimensional computational domain, which is occupied by three substances: the air, the fluid, and the solid body. The motion of the air is neglected and the effect of the ambient pressure exerted on fluid by air is taken into consideration. The motion of the body is predefined. Specifically, forced harmonic oscillation of a circular cylinder placed horizontally in uniform stream of a viscous incompressible fluid in the presence of a free surface is considered. The motion of the fluid is captured by computing flow parameters at a large but finite number of points (or knots) located within the computational domain. Each knot belongs to a computational cell. Cells are organized into a computational grid. The choice of computational grid is one of the major steps in determining the numerical approach to be employed. There are three main types of computational grids: structured/unstructured, boundary fitted/non-boundary fitted, and the Lagrangian/Eulerian grids.

A computational grid is called *structured* if it has a regular topology: all grid cells have the same number of faces, and all cell vertices are surrounded by the same number of cells. Each cell of a two-dimensional structured grid can be identified by two indices (i, j). A structured grid is called Cartesian if any two cell faces are either parallel or perpendicular to each other. All grids that are not structured are called unstructured grids. Structured, and specifically, Cartesian grids are easier to deal with, while unstructured grids allow more freedom in mesh refinement.

A grid is called *boundary fitted* if any part of fluid domain boundary coincides with cell faces of the grid. In a non-boundary fitted grid the fluid domain boundary passes through computational cells. The main advantage of using a boundary fitted grid is that discretization of boundary conditions is easy since grid cells are aligned with the fluid boundary. However, the generation of either a structured or an unstructured boundary fitted grid is a difficult task. In the presence of a free surface, the generation of a boundary fitted grid is required at each time iteration step, which is both difficult and time consuming. However, the use of a non-boundary fitted grid requires additional methods to track the fluid boundary as the fluid moves through the mesh.

A grid is called the Eulerian grid if all grid knots are fixed throughout the computation. In the Lagrangian grid, knots are treated as fluid particles carried by fluid flow. If the Lagrangian grid is of the boundary fitted type at the initial time, it will then remain boundary fitted throughout the simulation. However, fluid flow may lead to crowding of a large number of knots in certain locations as well as large fluid regions containing no knots at all. In such cases, a remeshing procedure is required, which leads to two major difficulties: (i) the determination of conditions under which the

remeshing procedure should be initiated; and (ii) the implementation of a specific remeshing algorithm.

Flow irregularities such as steep and breaking waves may occur at the free surface. Also, at low submergence depths, the cylinder may pierce the free surface, thus changing the fluid domain topology. Severe grid distortions make use of the Lagrangian grid impractical. The use of the Eulerian grid requires an interface tracking method to capture the exact location of the free surface which passes through computational cells. Two interface tracking methods are widely used: the level set method and the VOF method. In the level set method introduced by Osher and Sethian (1988), the interface is implicitly defined as a zero level set of the level set function ϕ. The interface is evolved by considering a numerical approximation of the following level set evolution equation on a fixed grid

$$\frac{\partial \phi}{\partial t} + \underline{u} \cdot \nabla \phi = 0. \tag{1.1}$$

A comprehensive review of the level set methods is provided in the work by Sethian and Smereka (2003). The VOF method introduced by Hirt and Nichols (1981) employs a discrete characteristic function which defines volume fractions occupied by the fluid within all computational cells to track the fluid domain. The exact free surface location is reconstructed and advected via geometrical procedures. A comprehensive review of the VOF method can be found in the works by Rider and Kothe (1998), Scardovelli and Zaleski (1999), Pilliod and Puckett (2004). Both the VOF and the level set methods are capable of handling large interface distortions and topology changes. However, the VOF method is much less prone to mass conservation errors

since it is based on geometrical procedures for interface reconstruction and advection. This makes the use of a structured Cartesian grid more practical for simulations of free surface flow problems.

A number of methods that treat rigid bodies of arbitrary geometry in Cartesian grids have been developed. In the fractional area/volume obstacle representation (FAVOR) method by Hirt and Sicilian (1985), curved boundaries are represented in rectangular grids with variable porosity which changes abruptly from unity to zero across a rigid boundary. The numerical representation of a variable porosity formulation is defined in terms of fractional areas and volumes open to flow, which are employed in numerical approximations of the governing equations. In the cut cell method by Udaykumar *et al.* (2001), the interface is tracked by marker particles and reconstructed using piecewise linear segments. Computational cells cut by the interface are reshaped by discarding the cell parts occupied by solid body. Small cells resulting from the reshaping procedure are merged with neighbouring cells thus forming trapezoidal control volumes. A complex two-dimensional polynomial interpolation function is used for accurate approximation of the mass, convective and diffusive fluxes in the finite volume formulation. In addition, a number of so-called diffuse interface methods have been developed. The immersed boundary method by Peskin (1977) tracks the body interface in a fixed Cartesian grid with a set of marker particles. The presence of the body is modeled with surface force incorporated into the governing equations. The interface between the fluid and the body has finite thickness due to smoothing of material properties across the boundary within a transition zone that is twice the grid step. In the fictitious domain method by Glowinski *et al.* (2001), moving rigid bodies are filled with surrounding fluid, which is then constrained to perform a rigid

body motion. Variational formulation involving the Lagrange multipliers is derived to match velocities of the fluid inside and outside of the body. Unlike the cut cell method and the FAVOR method, the fictitious domain method and the immersed boundary method model a rigid body with surface force and do not treat the body surface as a sharp interface. Since the interface between the fluid and the cylinder should be treated as a sharp interface for the problem under consideration, the methods by Hirt and Sicilian (1985) and Udaykumar *et al.* (2001) are more appropriate for numerical simulation of free surface flows past an oscillating cylinder.

A number of difficulties arise when combining the above mentioned methods with the VOF method. For example, a substantial modification of the cut cell method by Udaykumar *et al.* (2001) is developed in the works by Verstappen and Veldman (2000) and Gerrits (2001) to enable numerical simulation of free surface flows with stationary rigid bodies. The second order fractional time step method, employed by Udaykumar *et al.* (2001), is replaced with the first order explicit forward Euler method due to the fact that the region occupied with fluid changes with time. A staggered storage of fluid velocity components is used to avoid pressure decoupling problems and guarantees exact mass conservation. The finite volume discretization is done in such a way that there is no need to avoid small cells, so that no cell merging procedure is required. In addition, the modeling of rigid bodies moving through fixed computational grid poses great difficulties. For example, unavoidable numerical pressure spikes due to moving bodies are encountered in the recent works by Fekken (2004) and Kleefsman (2005) which are extensions of Gerrits' method for moving bodies.

In this thesis, the non-boundary fitted Eulerian grid of Cartesian type is used. This type of grid is easy to generate since it does not move with the fluid and does not require remeshing procedures. Fluid-air and fluid-body boundaries are tracked with the VOF and the FAVOR methods, respectively. The finite volume approximation of the governing equations is employed after the modification of the cut cell method by Gerrits (2001).

1.2 Governing equations

In the subsequent sections of this Chapter a new mathematical model for a class of flows including free surface flows with arbitrarily moving bodies is presented. First the governing equations are given and then the boundary conditions are stated.

The standard approach for obtaining a mathematical description of fluid flow is as follows (see, for example, Serrin (1959)). The fluid is treated as a continuum consisting of an infinite number of infinitesimal fluid particles. The evolution of an arbitrary material volume (i.e., the volume consisting of the same fluid particles at all times) is considered, and conservation laws are applied to it. The Reynolds transport theorem is then used to reformulate the conservation laws for a control volume which is fixed in space. This results in the integral form of the governing equations corresponding to a control volume. Numerical approximation of these integral equations corresponding to each computational cell results in a large system of equations to be solved computationally.

The above mentioned procedure considers volumes containing only fluid. It does not account for fluid interfaces. The use of the non-boundary fitted grid in this thesis makes passage of the fluid interface through computational cells possible. Thus, in order to apply conservation laws to all computational cells, the Reynolds transport theorem is extended for control volumes containing the fluid interface.

1.2.1 The Reynolds transport theorem

The detailed derivation of the well-known Reynolds transport theorem may be found, for example, in Serrin (1959). In this subsection a sketch of a typical application of this theorem is given. First, a computational cell V_{cv} is considered. At time t_0, this cell contains a material volume $V(t)$, which consists of all fluid particles located within V_{cv} at $t = t_0$. This material volume $V(t)$ does not change its mass with time, i.e.,

$$\frac{d}{dt}\int_{V(t)} \rho \, \mathrm{d}V = 0. \tag{1.2}$$

Here, the mass of material volume is obtained as an integral of fluid density ρ over the material volume $V(t)$. The Reynolds transport theorem is then used to express equation (1.2) in terms of the computational cell V_{cv} and its boundary S_{cv}

$$\frac{d}{dt}\int_{V(t)} \rho \, \mathrm{d}V = \frac{d}{dt}\int_{V_{\text{cv}}} \rho \, \mathrm{d}V + \int_{S_{\text{cv}}} \rho(\underline{u}\cdot\underline{n}) \, \mathrm{d}S. \tag{1.3}$$

Here, $\underline{u}$ is the velocity vector and $\underline{n}$ is the outward normal to the computational cell boundary. Thus, the application of the Reynolds transport theorem to the mass of

the material volume $V(t)$ results in equation (1.3), which states that the time rate of change of mass within the material volume is equal to the sum of the time rate of change of mass within the computational cell and the flux through the computational cell boundary.

1.2.2 Extension of the Reynolds transport theorem for control volumes containing a fluid interface

Equation (1.3) is not valid for control volumes that contain a fluid interface. However, it can be extended for this type of control volumes, and this extension plays a key role in the design of the numerical simulation tool developed in this thesis. In this section, a sketch of this derivation is given. The full details of this derivation will be reported elsewhere. This extension is based on the theory of generalized functions and is originally introduced in the work of Hirt and Sicilian (1985) to obtain a special form of the governing equations valid for inviscid fluid flows. These equations are called FAVOR equations and their technique is named as FAVOR method.

The majority of the previous theoretical studies conducted on generalized functions provide mathematically rigorous but extremely abstract representations, which can hardly be used in applications. Thus, engineers and scientists working with applications of generalized function theory sometimes employ various ad hoc methods, which are not mathematically strict but seem to work in practice. The link between the theory and applications of generalized functions is established by Farassat (1996), where utilization of generalized functions in applications is discussed using a rigor-

ous mathematical approach. Hirt and Sicilian (1985) did not clarify the generalized function properties used in the derivation of the FAVOR equations. This thesis uses the techniques discussed in Farassat (1996) to extend the Reynolds transport theorem for control volumes containing fluid interfaces. It is noted that the FAVOR approximation is a powerful and relatively simple modeling technique, yet it is not widely used. One of the advantages of the FAVOR method is that it can be combined with the VOF method for tracking the free surface in a straightforward manner. In general, volume-fraction methods are efficient provided that they are supplemented with special considerations for boundary conditions and for numerical stability (see, for example, Hirt (1992)). Hirt and his co-workers have applied the combined VOF-FAVOR method to a variety of free surface problems, including the problem of fluid slosh in cylindrical and spherical tanks, ship-wave resistance problems and the modeling of shallow flows over uneven surfaces, including wetting and drying phenomena (see Hirt and Sicilian (1985), Hirt (1992)). This combined method is based on techniques used in the commercial fluid dynamics simulation package FLOW-3D developed by Flow Science, Inc. This program is designed for unsteady, three-dimensional flows involving a free-surface and conforming to a complex geometry. Its strength as a robust analysis tool arises from the use of two special numerical techniques: VOF and FAVOR methods.

1.2.3 Generalized Reynolds transport theorem

A computational cell V_{cv} which is partly occupied with fluid is considered as shown in Figure 1.1. At $t = t_0$, it contains a material volume $V(t)$ which consists of all

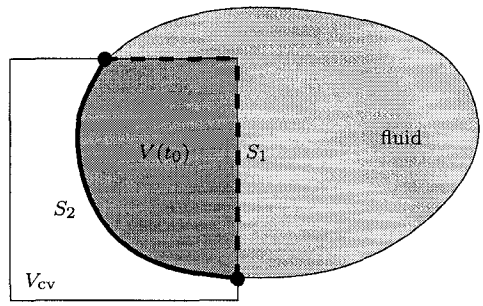

Figure 1.1: Control volume,V_{cv}, containing the fluid interface, S_2.

fluid particles located within V_{cv} at this specific time. The boundary of this material volume $\partial V(t)$ at $t = t_0$ can be divided into two parts: S_1 is the part which coincides with the control volume boundary, and S_2 is the fluid boundary located within the computational cell V_{cv}. Following the work of Farassat (1996), the level set representation of boundaries S_1 and S_2 is used. S_1 is associated with a level set function $f_1(\underline{x})$, which is equal to zero at all $\underline{x} \in S_1$, greater than zero within material volume $V(t)$ and less than zero otherwise. S_2 is associated with the level set function f_2. The level set functions are used to represent the mass of material volume $V(t)$ as an integral over the entire three dimensional space $\mathbb{R}$

$$\int_{V(t)} \rho \, \mathrm{d}V = \int_{\mathbb{R}} h(f_1)h(f_2)\rho \, \mathrm{d}V, \tag{1.4}$$

where $h(\xi)$ is the Heaviside step function

$$h(\xi) = \begin{cases} 1, & \xi > 0, \\ 0, & \text{otherwise.} \end{cases} \tag{1.5}$$

It is noted that fluid density, ρ, should also be analytically extended over the entire space in order to define the integral on the right hand side of equation (1.4). The time rate of change of the material volume mass can be then expressed by

$$\frac{d}{dt}\int\limits_{V(t)} \rho \, \mathrm{d}V = \frac{\overline{d}}{dt}\int\limits_{\mathbb{R}} h(f_1)h(f_2)\rho \, \mathrm{d}V, \tag{1.6}$$

where $\dfrac{\overline{d}}{dt}$ is the generalized time derivative which can be moved inside the integral based on generalized functions theory. Equation (1.6) can therefore be written as

$$\frac{d}{dt}\int\limits_{V(t)} \rho \, \mathrm{d}V = \int\limits_{\mathbb{R}} \left(\frac{\overline{d}(h(f_2)\rho)}{dt} \cdot h(f_1) + \frac{\overline{d}h(f_1)}{dt} \cdot h(f_2)\rho \right) \mathrm{d}V. \tag{1.7}$$

Following the works of Hirt and Sicilian (1985) and Farassat (1996), this equation can be written as

$$\frac{d}{dt}\int\limits_{V(t)} \rho \, \mathrm{d}V = \frac{d}{dt}\int\limits_{\mathbb{V}} \rho \, \mathrm{d}V + \int\limits_{\mathbb{A}} \rho \, (\underline{n} \cdot \underline{u}) \, \mathrm{d}S, \tag{1.8}$$

where $\mathbb{V}$ is the volume of the fluid located within computational cell V_{cv} at $t = t_0$ and $\mathbb{A}$ is the part of the computational cell boundary which is located in fluid. Equation (1.8) can be used to derive the mathematical description of the conservation of mass expressed in terms of fractional volumes $\mathbb{V}$ and fractional areas $\mathbb{A}$ open to flow. Similarly, the time rate of change of linear momentum of material volume $V(t)$ can be expressed as

$$\frac{d}{dt}\int\limits_{V(t)} \rho\underline{u} \, \mathrm{d}V = \frac{d}{dt}\int\limits_{\mathbb{V}} \rho\underline{u} \, \mathrm{d}V + \int\limits_{\mathbb{A}} \rho\underline{u} \, (\underline{n} \cdot \underline{u}) \, \mathrm{d}S. \tag{1.9}$$

Equations (1.8) and (1.9) form a basis for the derivation of equations governing the fluid flow which are suitable for both the computational grid and the numerical method employed in this thesis.

1.2.4 Derivation of the governing equations

It is important to note that equations (1.8) and (1.9) are valid for an arbitrary continuum. In this subsection, the governing equations describing the motion of a Newtonian incompressible viscous fluid are derived. The conservation of mass requires that the left hand side of equation (1.8) is equal to zero, i.e., the mass of fluid within V does not change. Consequently, equation (1.8) can be rewritten as

$$\frac{d}{dt}\int_{\mathbb{V}} \mathrm{d}V + \int_{\mathbb{A}} \underline{n}\cdot\underline{u}\,\mathrm{d}S = 0, \tag{1.10}$$

after making use of the fact that incompressible fluid density is constant.

The second governing equation is obtained by applying the Newton's second law of motion which states that the linear momentum of a material volume $V(t)$ changes due to external forces acting on it, i.e.,

$$\frac{d}{dt}\int_{V(t)} \rho\,\underline{u}\,\mathrm{d}V = \underline{F}. \tag{1.11}$$

There are two types of external forces which comprise $\underline{F}$: volumetric and surface forces. Volumetric forces act on the entire material volume, while surface forces affect

only a very thin area at the material volume boundary (see, for example, Serrin (1959))

$$\frac{d}{dt}\int\limits_{V(t)} \rho\underline{u}\,\mathrm{d}V = \int\limits_{V(t)} \rho\,\underline{g}\,\mathrm{d}V + \int\limits_{\partial V(t)} \underline{n}\cdot\underline{\underline{\sigma}}\,\mathrm{d}S. \tag{1.12}$$

Surface forces are determined using the stress tensor $\underline{\underline{\sigma}}$ of the form

$$\underline{\underline{\sigma}} = -p\,\underline{\underline{I}} + \mu\,\nabla\underline{u}. \tag{1.13}$$

Here, p is the fluid pressure, $\underline{\underline{I}}$ is the identity tensor, μ is the dynamic fluid viscosity, and $\mu\nabla\underline{u}$ is the viscous stress tensor. Substitution of (1.13) into (1.12) yields the integral form of the momentum equation

$$\frac{d}{dt}\int\limits_{V(t)} \rho\,\underline{u}\,\mathrm{d}V = -\int\limits_{\partial V(t)} p\,\underline{n}\,\mathrm{d}S + \mu\int\limits_{\partial V(t)} \underline{n}\cdot\nabla\underline{u}\,\mathrm{d}S + \int\limits_{V(t)} \rho\,\underline{g}\,\mathrm{d}V. \tag{1.14}$$

The left hand side of this equation can be written as

$$\frac{d}{dt}\int\limits_{\mathbb{V}} \rho\underline{u}\,\mathrm{d}V + \int\limits_{\mathbb{A}} \rho\underline{u}\,(\underline{n}\cdot\underline{u})\,\mathrm{d}S = -\int\limits_{\partial V(t)} p\underline{n}\,\mathrm{d}S + \mu\int\limits_{\partial V(t)} \underline{n}\cdot\nabla\underline{u}\,\mathrm{d}S + \int\limits_{V(t)} \rho\,\underline{g}\,\mathrm{d}V, \tag{1.15}$$

after making use of equation (1.9). It is noted that the boundary of the material volume, $\partial V(t)$, at $t = t_0$ consists of the fractional area, $\mathbb{A}$, open to flow, and the fluid interface, $\mathbb{I}$, located within the computational cell. Integrals of volumetric forces over material volume $V(t)$ can be replaced by integrals over the fractional volume $\mathbb{V}$.

Equation (1.15) can then be written as

$$\frac{d}{dt}\int_{\mathbb{V}} \underline{u}\,\mathrm{d}V + \int_{\mathbb{A}} \underline{u}\,(\underline{n}\cdot\underline{u})\,\mathrm{d}S = -\int_{\mathbb{A}\cup\mathbb{I}} p'\,\underline{n}\,\mathrm{d}S + \nu\int_{\mathbb{A}\cup\mathbb{I}} \underline{n}\cdot\nabla\underline{u}\,\mathrm{d}S + \int_{\mathbb{V}} \underline{g}\,\mathrm{d}V, \qquad (1.16)$$

after eliminating the constant fluid density, ρ. Here, $\nu = \mu/\rho$ is the kinematic fluid viscosity and $p' = p/\rho$. Equations (1.10) and (1.16) are the integral form of the Navier-Stokes equations describing the motion of a Newtonian incompressible fluid.

1.3 Boundary conditions

In this section, boundary conditions are stated at the solid boundary and at the free surface.

1.3.1 Solid body conditions

At the solid boundary, the no-slip boundary condition for a viscous fluid

$$\underline{u} = \underline{u}_{\text{body}} \qquad (1.17)$$

is applied, which states that the solid boundary is impermeable and the flow sticks to the wall due to viscous effects.

1.3.2 Free surface conditions

Free surface boundary conditions for the velocity and the pressure result from balancing the surface forces on the contact line between the air and the fluid (see, for example, Rood (1995)). The tangential free surface condition

$$\frac{\partial u_n}{\partial \underline{\tau}} + \frac{\partial u_\tau}{\partial \underline{n}} = 0 \tag{1.18}$$

states that the air exerts no tangential stress on the fluid since the motion of the air is neglected throughout this thesis. The normal surface force condition is given by

$$2\nu \frac{\partial u_n}{\partial \underline{n}} - p = -p_0. \tag{1.19}$$

In these equations u_n and u_τ are the normal and tangential fluid velocity components, respectively. Equation (1.19) states that the air exerts ambient pressure p_0 on the fluid. It is noted that surface tension is neglected since a class of flows in which gravitational force plays a more important role than the surface tension is considered.

1.4 Problem specification

In the present study, the mathematical model introduced in the previous section is applied to the problem of unsteady, two-dimensional, laminar uniform free surface flow past a transversely oscillating circular cylinder as shown in Figure 1.2. The upstream uniform flow velocity is U and the cylinder is submerged a distance, h^*,

below the position of the undisturbed horizontal free surface. The diameter of the cylinder is d. The characteristic length and velocity scales are taken to be the cylinder

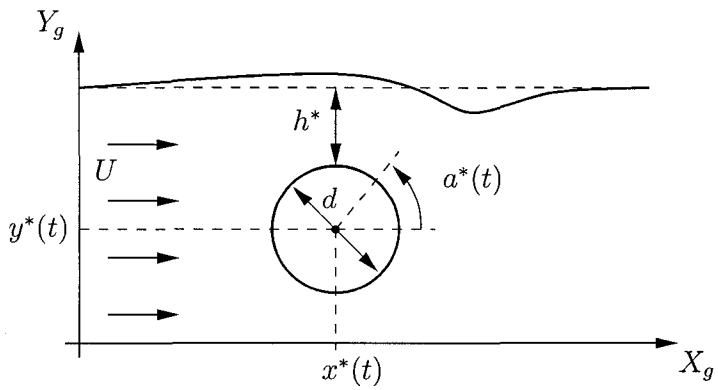

Figure 1.2: Sketch of the problem.

diameter, d, and free stream velocity, U, respectively. The dimensionless parameters are the Reynolds number

$$\mathrm{R} = \frac{Ud}{\nu}, \tag{1.20}$$

where ν is the kinematic viscosity of the fluid; the amplitude of the cylinder oscillation,

$$A = \frac{A^*}{d}, \tag{1.21}$$

the frequency ratio, f/f_0, with

$$f = \frac{df^*}{U}, \tag{1.22}$$

and

$$f_0 = \frac{df_0^*}{U} \tag{1.23}$$

being the dimensionless forcing frequency of the cylinder oscillation and the natural vortex shedding frequency of a stationary cylinder, respectively; the submergence

depth

$$h = \frac{h^*}{d}, \tag{1.24}$$

and the Froude number

$$\mathrm{Fr} = \frac{U}{\sqrt{dg}}, \tag{1.25}$$

where g is the acceleration due to gravity. Throughout this thesis the specific values $d = 1$ and $U = 1$ are used for simplicity. Thus, $\mathrm{R} = 1/\nu$ and $\mathrm{Fr} = 1/\sqrt{g}$. It is noted that due to the presence of a free surface the acceleration due to gravity exerts an influence and must be considered. Here, A^* is the dimensional amplitude of the cylinder oscillation, f^* is the dimensional forcing frequency of an oscillating cylinder, f_0^* is the dimensional natural vortex shedding frequency of a stationary cylinder, and h^* is the dimensional submergence depth.

The cylinder is initially at rest in a uniform flow with velocity U and then, at time $t = 0$, the cylinder starts to perform a predefined motion which is described by means of three dimensionless generalized coordinates

$$x(t) = \frac{x^*(t)}{d}, \qquad y(t) = \frac{y^*(t)}{d}, \qquad a(t) \equiv a^*(t), \tag{1.26}$$

so that at $t = t_0$, the center of the cylinder cross-section is located at $(x_0, y_0, 0) = (x(t_0), y(t_0), 0)$, and the angular displacement of the cylinder is $a_0 = a(t_0)$ as shown in Figure 1.2. The dimensionless cylinder velocity $\underline{u}_{\mathrm{body}}$ at any point $(x, y, 0)$ on the cylinder boundary, which is required to apply the no-slip boundary condition,

can be then determined as follows

$$\underline{u}_{\text{body}} = (\dot{x} - \dot{a}(y - y_0),\ \dot{y} + \dot{a}(x - x_0), 0). \tag{1.27}$$

Thus, appropriate choices for $x(t)$, $y(t)$ and $a(t)$ define the cylinder motion. For example, the following specific choices for these coordinates

$$x(t) = 0, \qquad y(t) = -A\cos(2\pi ft), \qquad a(t) = 0 \tag{1.28}$$

define the time dependent motion for the case of a transversely oscillating cylinder so that $\underline{u}_{\text{body}} = (0, 2A\pi ft\sin(2\pi ft), 0)$. Here, t is the dimensionless time defined in terms of the dimensional time, t^*, as

$$t = \frac{Ut^*}{d}. \tag{1.29}$$

The dimensionless period of the cylinder oscillation, T, is defined as

$$T = \frac{1}{f}. \tag{1.30}$$

For transverse oscillations the range near $f/f_0 \approx 1.0$ constitutes the fundamental *lock-on* regime.

The dimensionless force exerted by the cylinder on the fluid,

$$\underline{F} = \underline{F}^*/(\frac{1}{2}\rho dU^2), \tag{1.31}$$

has two components, C_D and C_L, the dimensionless drag and lift coefficients, respectively. They are defined as

$$C_D = C_{D_P} + C_{D_V}, \qquad C_L = C_{L_P} + C_{L_V}, \tag{1.32}$$

where C_{D_P} and C_{L_P} are the contributions due to the pressure gradient, and C_{D_V} and C_{L_V} are the contributions due to the viscous shear forces. The pressure and viscous contributions are computed as

$$C_{D_P} = \int_0^{2\pi} p \cos\theta \, \mathrm{d}\theta, \qquad C_{L_P} = \int_0^{2\pi} p \sin\theta \, \mathrm{d}\theta, \tag{1.33}$$

$$C_{D_V} = \frac{1}{\mathrm{R}} \int_0^{2\pi} \frac{\partial u}{\partial \underline{n}} \, \mathrm{d}\theta, \qquad C_{L_V} = \frac{1}{\mathrm{R}} \int_0^{2\pi} \frac{\partial v}{\partial \underline{n}} \, \mathrm{d}\theta. \tag{1.34}$$

Here, $\underline{n} = (\cos(\theta), \sin(\theta), 0)$ is the outward unit normal to the cylinder boundary.

Two different techniques can be employed to incorporate the motion of the cylinder into the computational model: (i) the cylinder moves through a stationary computational grid; (ii) the cylinder is stationary with respect to the grid which moves with respect to an inertial frame of reference. In this work, both methods are enabled by considering three reference frames: $\mathbb{X}$ is the inertial reference frame, $\mathbb{X}_g$ is the reference frame in which the computational grid is defined, and $\mathbb{X}_c$ is the reference frame that moves and rotates with the cylinder. $\mathbb{X}_g$ is restricted to move without rotation, so its axes are always parallel to the axes of the inertial system. The cylinder motion (1.28) is specified with respect to the grid reference frame $\mathbb{X}_g$. Thus, the equations of motion of the grid reference frame $\mathbb{X}_g$ with respect to the inertial reference frame $\mathbb{X}$

are provided to complete the specification of the cylinder motion

$$x_g = x_g(t), \qquad y_g = y_g(t). \tag{1.35}$$

In general, $\mathbb{X}_g$ is not inertial. As it is shown in Appendix A, this can be accounted for by modifying the gravitational acceleration $\underline{g_0} = (0, g_0, 0)$

$$\underline{g} = (-\ddot{x}_g(t),\ g_0 - \ddot{y}_g(t), 0). \tag{1.36}$$

For example, by choosing $x(t) = 0$, $y(t) = 0$, $a(t) = 0$, $x_g = 0$, $y_g = -A\cos(2\pi ft)$, the problem set-up corresponding to a cylinder which is stationary with respect to a transversely oscillating grid can be obtained, which is equivalent to the problem of a transversely oscillating cylinder with respect to the inertial frame of reference. In summary, the governing equations and boundary conditions for the problem under consideration are

$$\frac{d}{dt}\int_{\mathbb{V}} \mathrm{d}V + \int_{\mathbb{A}} \underline{n}\cdot\underline{u}\,\mathrm{d}S = 0, \tag{1.37a}$$

$$\frac{d}{dt}\int_{\mathbb{V}} u\,\mathrm{d}V + \int_{\mathbb{A}} u\,(\underline{n}\cdot\underline{u})\,\mathrm{d}S = -\int_{\mathbb{A}\cup\mathbb{I}} p\,n_x\,\mathrm{d}S + \frac{1}{\mathrm{R}}\int_{\mathbb{A}\cup\mathbb{I}} \underline{n}\cdot\nabla u\,\mathrm{d}S, \tag{1.37b}$$

$$\frac{d}{dt}\int_{\mathbb{V}} v\,\mathrm{d}V + \int_{\mathbb{A}} v\,(\underline{n}\cdot\underline{u})\,\mathrm{d}S = -\int_{\mathbb{A}\cup\mathbb{I}} p\,n_y\,\mathrm{d}S + \frac{1}{\mathrm{R}}\int_{\mathbb{A}\cup\mathbb{I}} \underline{n}\cdot\nabla v\,\mathrm{d}S \tag{1.37c}$$

$$+\int_{\mathbb{V}} \left(\frac{1}{\mathrm{Fr}^2} - 4A\pi^2 f^2\cos(2\pi ft)\right)\mathrm{d}V, \tag{1.37d}$$

and

$$u = 0, \quad v = 0 \qquad \text{at the cylinder boundary,} \tag{1.37e}$$

$$\frac{\partial u_n}{\partial \underline{\tau}} + \frac{\partial u_\tau}{\partial \underline{n}} = 0 \qquad \text{at the free surface,} \tag{1.37f}$$

$$\frac{2}{\mathrm{R}} \frac{\partial u_n}{\partial \underline{n}} - p = -p_0 \qquad \text{at the free surface.} \tag{1.37g}$$

Chapter 2

Numerical model

The theoretical development of fluid dynamics focuses on the construction and solution of the governing equations for different categories of fluid dynamics and the study of various approximations to those equations. The steady improvement in the speed of computers and the memory size since the early 1950s has led to the emergence of computational fluid dynamics (CFD). This branch of fluid dynamics complements experimental and theoretical fluid dynamics by providing an alternative cost-effective means of simulating real flows. As it can be seen from the previous Chapter, flows and related phenomena can be described by partial differential (or integro-differential) equations, which cannot be solved analytically except in special cases. Much as the accuracy of experimental data depends on the quality of the tools used, the accuracy of computational methods depends on the quality of the discretizations used. In the previous Chapter the basic governing equations are presented in a special integral form, which serves as a starting point for an important class of numerical methods.

There are three well-known discretization approaches: finite difference, finite element, and finite volume methods. In finite difference methods, the derivatives appearing in the governing equations are approximated with finite differences, which are the discrete analog of the derivatives. Finite difference methods are mostly used on structured and especially Cartesian grids, where finite differences can be derived by Taylor-series expansions. The literature on finite difference methods in computational fluid dynamics is adequately documented in many books such as Patankar (1980), Fletcher (1988) and Anderson (1995). In finite element methods, the computational domain is divided into discrete volumes called finite elements. The solution is approximated with a linear combination of *basis functions* defined for every element. This approximation is substituted into the *weak form* of the governing equations which is obtained by multiplying the governing equations with *test functions* and integrating by parts, resulting in a set of linear equations for the solution. The book by Gresho and Sani (1998) is dedicated to the application of finite element methods to the incompressible fluid flow problems. Other literature on finite element methods is referenced in the work by Chung (2002). Finite volume methods apply conservation principles for every computational cell (control volume) by computing convective, diffusive and mass fluxes through cell faces. Physical quantities such as mass and momentum are conserved by choosing control volumes in such a way that every cell face belongs to exactly two cells, so that fluxes that leave one control volume are gained in adjacent ones. Comprehensive information on finite volume methods can be found, for example, in the books by Versteeg and Malalasekera (1995) and Lomax *et al.* (2001).

Finite difference methods are straightforward and very effective on structured grids.

However, for unstructured grids such methods are prohibitive, and curved grid lines must be transformed into orthogonal coordinates so that the finite difference expressions can be formulated. This restricts finite difference methods to simple geometries. On the contrary, complex geometries and unstructured grids are easily accommodated in finite element and finite volume methods without coordinate transformations. The underlying principles and formulations of finite element methods involve intricate mathematical analysis, which makes these methods harder to apply than finite difference and finite volume methods. Finite volume methods involve simple underlying principles based on direct invocation of conservation laws and at the same time provide the geometrical flexibility of finite element methods.

In this thesis, a Cartesian non-boundary fitted Eulerian grid is used. In the previous Chapter, the governing equations are represented in a form which is suitable for finite volume discretization. Boundary conditions are stated at the solid boundary and at the free surface in Section 1.3. Well-posed boundary conditions are enforced at the inflow and outflow boundaries to ensure the correct physical development of the flow near computational flow boundaries.

2.1 Computational grid

The motion of the fluid is determined by computing fluid velocity and pressure at certain discrete locations (knots) on each time iteration. A finite volume discretization is applied to computational cells (control volumes) which are built around these knots. It is noted that for incompressible fluids, only the pressure, p, and two components

of the velocity vector, $\underline{u} = (u, v, 0)$, are required to describe the flow. In the presence of a free surface, the VOF function, F, is needed to describe the configuration of the fluid domain as explained in Section 2.2.

In this thesis, a staggered grid is used. In Cartesian coordinates, the staggered arrangement introduced by Harlow and Welsh (1965) has several advantages over the collocated arrangement. For example, it prevents pressure oscillations observed on collocated grids since the approximation of pressure gradients on this type of grids includes the pressure difference between two *adjacent* grid knots, and not between *alternate* ones. In a full staggered arrangement, velocity knots are located on the edges of computational cells corresponding to pressure knots as shown in Figure 2.1. The

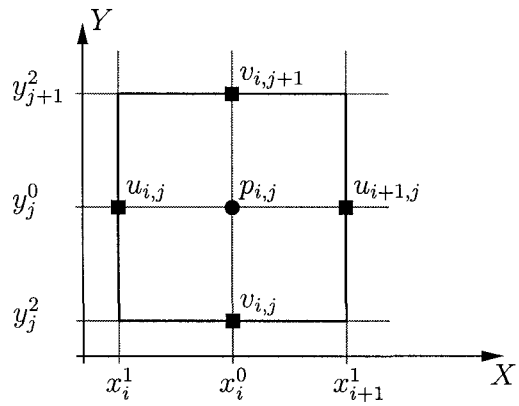

Figure 2.1: Staggered arrangement of pressure/velocity knots.

staggered grid involves three different sets of knots: pressure knots $p_{i,j}$, X-velocity knots $u_{i,j}$, and Y-velocity knots $v_{i,j}$. Specifically, the knot coordinates are defined as

$$p_{i,j} = \left(x_i^0, y_j^0, 0\right), \qquad u_{i,j} = \left(x_i^1, y_j^1, 0\right), \qquad v_{i,j} = \left(x_i^2, y_j^2, 0\right). \tag{2.1}$$

Indices i and j run along the X and Y axis, respectively, and $i = 0 \ldots N-1$, $j = 0 \ldots M-1$. If there are $N \times M$ pressure knots, there will be $(N+1) \times M$ u-velocity knots and $N \times (M+1)$ v-velocity knots. The velocity knot coordinates are computed as follows

$$x_i^1 = \frac{x_{i-1}^0 + x_i^0}{2}, \quad y_j^1 = y_j^0, \quad x_i^2 = x_i^0, \quad y_j^2 = \frac{y_{j-1}^0 + y_j^0}{2}. \tag{2.2}$$

Here, x_{-1}^0 and y_{-1}^0 are the ghost knots with coordinates such that $(x_1^0 - x_0^0) = (x_0^0 - x_{-1}^0)$ and $(y_1^0 - y_0^0) = (y_0^0 - y_{-1}^0)$.

Three types of control volumes corresponding to knots $p_{i,j}$, $u_{i,j}$ and $v_{i,j}$ are shown in Figure 2.2. Control volumes $P(i,j)$ define the primary computational cells used throughout this thesis.

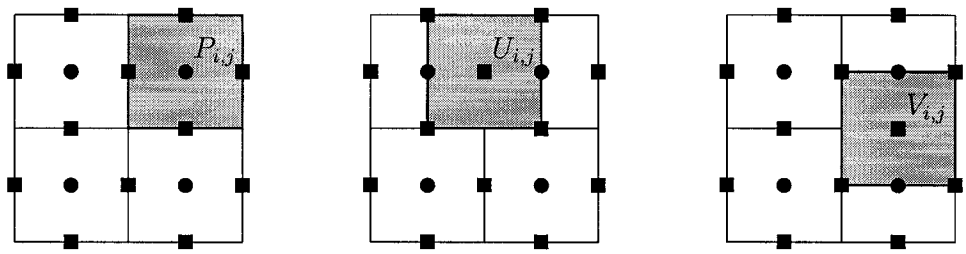

Figure 2.2: Control volumes corresponding to the pressure (left), u-velocity (middle), and v-velocity (right) knots.

2.2 Tracking the fluid domain

The numerical approach used in this work requires a special algorithm for tracking the fluid region within the computational grid. This is done by using the VOF method

due to Hirt and Nichols (1981). This method consists of two parts. First, the fluid boundary is reconstructed using the VOF function, $0 \leqslant F \leqslant 1$, which denotes the fractional volume of a computational cell (fluid fraction) that is occupied by the fluid. For example, $F_{i,j} = 1$ corresponds to the computational cell $P(i,j)$ which is entirely filled with fluid and $F_{i,j} = 0$ means that there is no fluid in the computational cell. Second, the fluid region is advected in time to its new position with the fluid velocity field based on the fluid domain configuration reconstructed at the previous time step.

2.2.1 Fluid interface reconstruction

Computational cells that contain both the fluid and the air are called interface cells (mixed cells). The exact free surface location in interface cells can be reconstructed from the values of the VOF function F. In each mixed cell, the fluid interface is approximated with a straight line segment. The equation of the line is specified by its unit outward normal $\underline{n} = (n_x, n_y, 0)$ and the distance $\bar{d}$ from the coordinate system origin

$$\underline{n} \cdot \underline{x} + \bar{d} = 0. \tag{2.3}$$

First, $\underline{n}$ is approximated and constant $\bar{d}$ is then computed so that the interface line should cut out exactly the same amount of fluid within the computational cell $P_{i,j}$ as specified by the VOF function $F_{i,j}$. The Brent's root finding algorithm is used to compute $\bar{d}$ (see Rider and Kothe (1998)).

The algorithm for finding the interface normal, $\underline{n}$, is not unique. The basic idea for estimating $\underline{n}$ is based on the values of F in the neighbourhood of the cell under

consideration. Several estimation algorithms can be found in a comprehensive study by Pilliod and Puckett (2004). In the present work, the algorithm due to Gerrits (2001) is used, in which the analytical interface normal

$$\underline{n} = \frac{\nabla F}{|\nabla F|} \tag{2.4}$$

is approximated by discretizing equation (2.4) with respect to all the neighbours of the interface cell. This results in a 2×2 linear system for $\underline{n} = (n_x, n_y, 0)$ which is then solved using Cramer's method.

2.2.2 Fluid advection

The fluid region is advected with the fluid velocity field at each time step. The advection algorithm computes the new values, F^{k+1}, of the VOF function in all computational cells based on the old values, F^k, and the fluid velocity field $\underline{u}^k = (u^k, v^k, 0)$. Here, superscripts k and $k+1$ correspond to the time step numbers and $t^{k+1} = t^k + \Delta t$, where Δt is the time step.

A number of different advection algorithms exists (see, for example, Hirt and Nichols (1981), Rider and Kothe (1998), Aulisa *et al.* (2003), Pilliod and Puckett (2004)). In this thesis, the area-preserving advection method due to Aulisa *et al.* (2003) is adopted, where an area-preserving linear mapping Π_{xy} is used to transform the fluid region at time step k, t^k, into the fluid region at time t^{k+1}. Π_{xy} is a combination of two linear mappings Π_x and Π_y along the X and the Y directions, respectively.

The first mapping, $\Pi_x : x' \rightarrow x$, maps the rectangle $A'B'C'D'$ onto the computational cell $P(i,j)$ which is the rectangle $ABCD$ (see Figure 2.3), where the X-coordinates of computational cell corners A and B are x_1 and x_2, respectively, and the X-coordinates of points A' and B' are $x_{\text{left}} = x_1 - u_{i,j}\Delta t$ and $x_{\text{right}} = x_2 - u_{i+1,j}\Delta t$, respectively. The first linear mapping Π_x performs a linear transformation of coordinates x' into

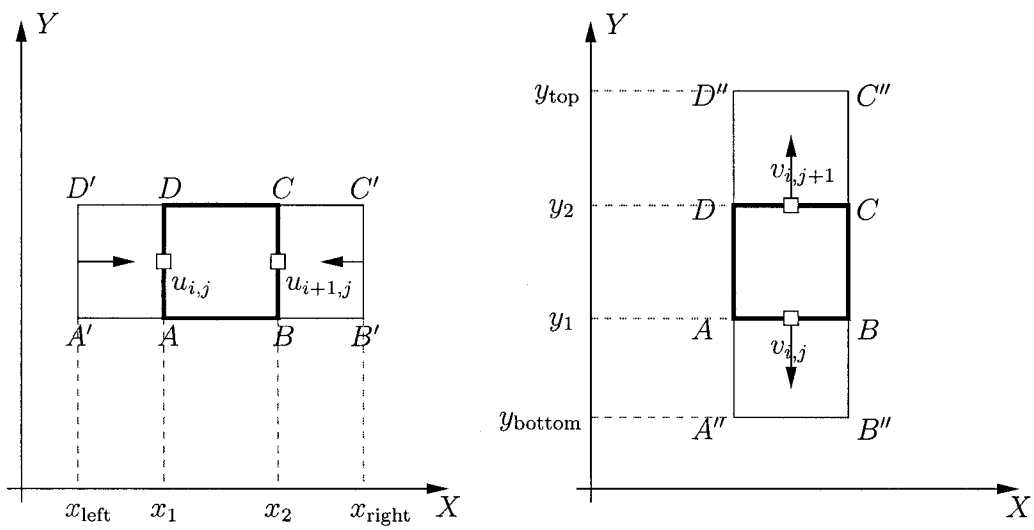

Figure 2.3: The linear mappings Π_x and Π_y: Π_x maps the rectangle $A'B'C'D'$ onto $ABCD$ (left) and Π'_y maps the rectangle $ABCD$ onto $A''B''C''D''$ (right).

x according to the following rule

$$x = ax' + b, \quad \text{where} \quad a = \frac{x_2 - x_1}{x_{\text{right}} - x_{\text{left}}} \quad \text{and} \quad b = x_1 - ax_{\text{left}}. \tag{2.5}$$

The second linear mapping $\Pi_y : y \rightarrow y''$ maps rectangle $ABCD$ onto $A''B''C''D''$, where the Y-coordinates of computational cell corners A and D are y_1 and y_2, respectively, and the Y-coordinates of points A'' and D'' are $y_{\text{bottom}} = y_1 + v_{i,j}\Delta t$ and $y_{\text{top}} = y_2 + v_{i,j+1}\Delta t$, respectively. Π_y performs a linear transformation of coordinates

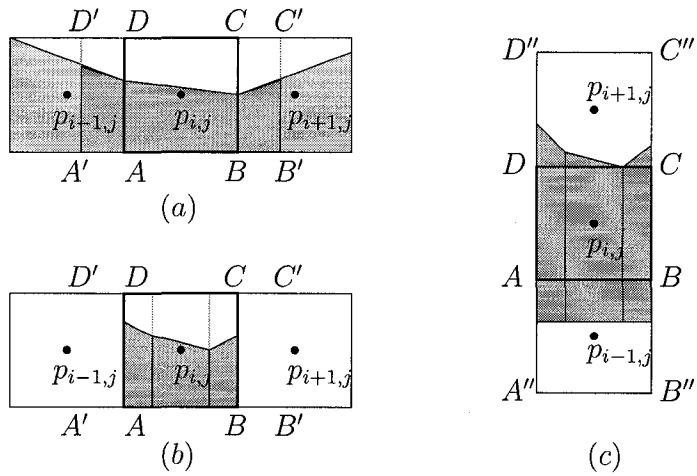

Figure 2.4: The fluid advection: (a) fluid polygons contained in the rectangle $A'B'C'D'$; (b) fluid polygons after applying the first mapping Π_x; (c) fluid polygons after applying the second mapping Π_y.

y into y'' according to the following rule

$$y'' = cy + d, \quad \text{where} \quad c = \frac{y_{\text{top}} - y_{\text{bottom}}}{y_2 - y_1} \quad \text{and} \quad d = y_{\text{bottom}} - cy_1. \tag{2.6}$$

The linear mapping, Π_{xy}, is a combination of the two mappings Π_x and Π_y. In the work by Aulisa *et al.* (2003), it is shown that this mapping preserves the area, provided that the fluid velocity field is divergence free. Thus, the fluid volume and mass are conserved exactly when using this advection algorithm in two dimensions.

The fluid advection process consists of three steps which are shown in Figure 2.4. First, fluid polygons located within the rectangle $A'B'C'D'$ are identified. The fluid from three computational cells $P(i-1,j)$, $P(i,j)$ and $P(i+1,j)$ may contribute to these polygons. Next, mappings Π_x and Π_y are applied to each of these fluid polygons. The images of fluid polygons shown in Figure 2.4 (c) are intersected with all eight

neighbours of the cell $P(i,j)$ as well as the cell $P(i,j)$ itself. The normalized areas of the intersections contribute to the values of the VOF function at time step $k+1$, F^{k+1}, corresponding to these cells. This procedure is repeated for each computational cell.

2.3 Classification of the pressure cells and the velocity knots

The unknown values of the pressure $p_{i,j}^{k+1}$ and the two velocity components $u_{i,j}^{k+1}$, $v_{i,j}^{k+1}$ at t^{k+1} are approximated by solving a linear system. A linear equation for $p_{i,j}^{k+1}$, $u_{i,j}^{k+1}$, $v_{i,j}^{k+1}$ is associated with each computational knot.

Since the non-boundary fitted grid is used in this thesis, the computational knots may be located either inside the fluid, the air, the solid, or at the boundary of the computational domain. All of the knots are categorized into classes, and linear equations of different types are associated with each computational knot, depending on which class the knot belongs to. Following the classification of Gerrits (2001), all of the knots are divided into the six classes listed in Table 2.1 and illustrated in Figure 2.5.

The classification of the pressure/velocity knots induces the classification of the pressure/velocity cells due to a one-to-one correspondence between the cells and the knots. In this thesis, the classification of the pressure knots/cells is referred to as the *classification of cells* whereas the classification of the velocity knots/cells is referred to as

Type	Cell/Knot location
E	in the air
F	in the fluid
X	in the solid
S	at the fluid-air interface
B	at the fluid-solid interface
D	at the computational domain boundary

Table 2.1: Classification of the pressure and the velocity knots.

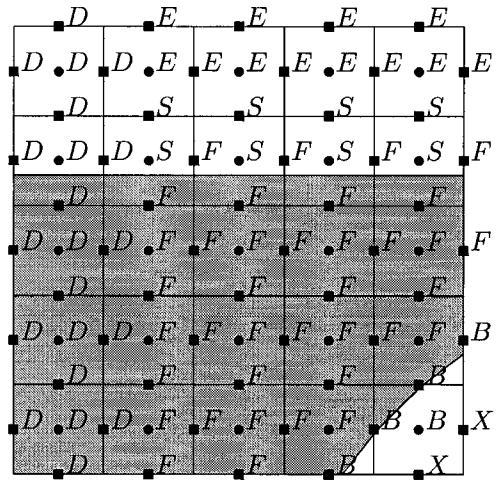

Figure 2.5: An example of the pressure and the velocity knot labeling. The fluid domain is shaded.

the *classification of knots* in order to distinguish these classifications.

In the cells/knots classification, the geometry of the solid body is accounted for by the volume of solid function, $0 \leqslant S \leqslant 1$, which denotes the fractional volume of a computational cell (solid fraction) that is occupied by the solid body. For example, $S_{i,j} = 1$ corresponds to the computational cell $P(i,j)$ which is entirely filled with solid and $S_{i,j} = 0$ means that there is no solid in the computational cell. The classification of the pressure cells uses the values of the VOF function $F_{i,j}$ as well as the values of the volume of solid function $S_{i,j}$ corresponding to each computational cell $P(i,j)$. The fluid fractions are known due to the VOF reconstruction and advection, as discussed in Section 2.2. On the other hand, solid fractions $S_{i,j}$ are considered to be known due to the specified solid body geometry.

The classification of the pressure cells is performed in three steps. In the first step,

the solid, the air and the fluid cells are assigned labels. All of the cells with $S_{i,j} > 0.5$ are marked as X-cells. All of the cells which are not X-cells are then marked either E-cells, if $F_{i,j} = 0$, or F-cells, otherwise. In the second step, cells located near fluid-air and fluid-solid interfaces are labeled. Every F-cell that has at least one neighbour cell which is E-cell, is marked as a S-cell. Every X-cell that has at least one neighbour which is F-cell, is marked as a B-cell. In the third step, labels are assigned to the pressure cells located at the boundary of the computational domain. Each cell $P(i, j)$ with either $i = 0$ or $j = 0$, or $i = N - 1$, or $j = M - 1$ is marked as a D-cell.

The classification of the velocity knots is performed as follows. Every u-velocity knot $u_{i,j}$ is located on the vertical edge shared by the computational cells $P(i - 1, j)$ and $P(i, j)$, while every v-velocity knot $v_{i,j}$ is located on the horizontal edge shared by the computational cells $P(i, j - 1)$ and $P(i, j)$. Thus, each velocity knot is assigned a label based on the labels of two pressure knots corresponding to these computational cells. Initially, all of the velocity knots are marked as F-knots. Then, the velocity knot located between two E-cells is marked as an E-knot. Velocity knots located between an E-cell and a F-cell are marked as S-knots. If at least one of the two cells surrounding a velocity knot is a X-cell, then this velocity knot is marked as a X-knot. If at least one of the two cells surrounding a velocity knot is a B-cell, then this velocity knot is marked as a B-knot. Finally, all the velocity knots residing on edges of D-cells are marked as D-knots.

The various types of linear equations added for each type of pressure cell/velocity knot are summarized in Section 2.8.

2.4 Discretization of the governing equations

A distinct method of obtaining a linear equation associated with cells/knots of this type is introduced for each pressure cell and velocity knot type. In this thesis, these methods are called *discretization patterns*. In what follows, discretization patterns corresponding to all types of cells and knots are described.

2.4.1 Computation of fractional areas/volumes open to flow

Equation (1.10),

$$\frac{d}{dt}\int_{\mathbb{V}} \mathrm{d}V + \int_{\mathbb{A}} \underline{n}\cdot\underline{u}\,\mathrm{d}S = 0,$$

represents the integral form of conservation of mass. It is applied to the computational cell $P(i,j)$ as shown in Figure 2.6. In equation (1.10), $\mathbb{V}$ is approximated by the

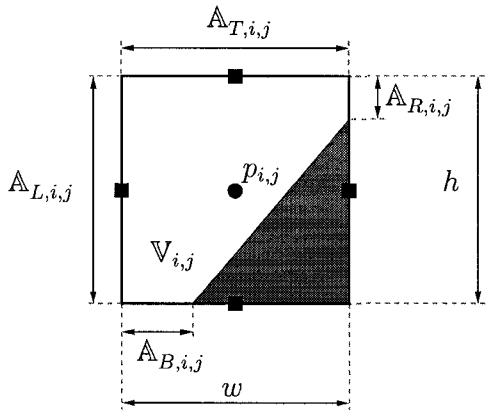

Figure 2.6: Fractional areas and volume open to flow corresponding to the computational cell $P(i,j)$. Solid body is shaded.

fractional volume, $\mathbb{V}_{i,j}$, open to flow in the cell $P(i,j)$, which is computed from

the value of the volume of solid function $S_{i,j}$ and the dimensions w and h of the computational cell $P(i,j)$ using

$$\mathbb{V}_{i,j} = (1 - S_{i,j})\, w\, h. \tag{2.7}$$

In equation (1.10), $\mathbb{A}$ is split into four parts corresponding to the left, right, top and bottom edges of the computational cell $P(i,j)$. Fractional areas $\mathbb{A}_{L,i,j}$, $\mathbb{A}_{R,i,j}$, $\mathbb{A}_{T,i,j}$ and $\mathbb{A}_{B,i,j}$ are computed by intersecting the edges of the cell $P(i,j)$ with the solid body. In this thesis fractional areas and volumes are not normalized with respect to total areas and volumes of control volume, unlike the studies by Gerrits (2001) and Hirt and Nichols (1981). This simplifies the discretization of the governing equations since the edge and volume dimensions are already incorporated into $\mathbb{A}_{i,j}$ and $\mathbb{V}_{i,j}$. In the case when the solid body moves through the computational grid, two sets of fractional areas and volumes are considered at each time step: $\mathbb{V}^k$ and $\mathbb{A}^k$ which correspond to time t^k, and $\mathbb{V}^{k+1}$ and $\mathbb{A}^{k+1}$ which correspond to time t^{k+1}.

2.4.2 The continuity equation

The computational cell shown in Figure 2.6 is considered as a control volume for the discretization of the continuity equation (1.10),

$$\frac{d}{dt}\int_{\mathbb{V}} \mathrm{d}V + \int_{\mathbb{A}} \underline{n} \cdot \underline{u}\, \mathrm{d}S = 0,$$

which consists of the unsteady and convective terms. The unsteady term is the time derivative of the fractional volume open to flow $\mathbb{V}_{i,j}$. This term is approximated as follows

$$\frac{d}{dt}\int_{\mathbb{V}} \mathrm{d}V \approx \frac{\mathbb{V}_{i,j}^{k+1} - \mathbb{V}_{i,j}^{k}}{\Delta t}. \tag{2.8}$$

The convective term is split into four integrals along the fractional areas. Since the edges of the computational cell $P(i,j)$ are either vertical or horizontal, the scalar product $\underline{u}\cdot\underline{n}$ can be calculated for each edge. This leads to the following approximation of the convective term

$$\int_{\mathbb{A}} (\underline{n}\cdot\underline{u})\,\mathrm{d}S \approx -\mathbb{A}_{L,i,j}^{k+1}\,u_{i,j}^{k+1} + \mathbb{A}_{R,i,j}^{k+1}\,u_{i+1,j}^{k+1} - \mathbb{A}_{B,i,j}^{k+1}\,v_{i,j}^{k+1} + \mathbb{A}_{T,i,j}^{k+1}\,v_{i,j+1}^{k+1}. \tag{2.9}$$

Substitution of (2.8) and (2.9) into (1.10) yields the following linear equation

$$\frac{\mathbb{V}_{i,j}^{k+1} - \mathbb{V}_{i,j}^{k}}{\Delta t} - \mathbb{A}_{L,i,j}^{k+1}\,u_{i,j}^{k+1} + \mathbb{A}_{R,i,j}^{k+1}\,u_{i+1,j}^{k+1} - \mathbb{A}_{B,i,j}^{k+1}\,v_{i,j}^{k+1} + \mathbb{A}_{T,i,j}^{k+1}\,v_{i,j+1}^{k+1} = 0. \tag{2.10}$$

On the right hand side of this equation, the sum of contributions made by the unsteady and convective terms appears. There are four unknowns in equation (2.10): $u_{i,j}^{k+1}$, $u_{i+1,j}^{k+1}$, $v_{i,j}^{k+1}$, and $v_{i,j+1}^{k+1}$. The values of the fractional areas and volumes for both times t^k and t^{k+1} are to be computed beforehand. This equation is a numerical counterpart of the continuity equation (1.10) applied to $P(i,j)$.

All of the terms free of unknowns can be moved to the right hand side of equation (2.10). However, this is not done with the intention of keeping the discretization of different terms separate. The following process is used for implementing this in a

computer program. The discretization is started with the trivial linear equation $0 = 0$. Discretization of each term adds contributions to the left hand side of this equation. After discretization of the last term is complete, the equation is ready to be added to the linear system under construction. With this approach, large equations with explicit summation of contributions made by all the terms, like (2.10), are never used.

2.4.3 The momentum equation

Equation (1.16),

$$\frac{d}{dt}\int_{\mathbb{V}} \underline{u}\,\mathrm{d}V + \int_{\mathbb{A}} \underline{u}\,(\underline{n}\cdot\underline{u})\,\mathrm{d}S = -\int_{\mathbb{A}\cup\mathbb{I}} p\,\underline{n}\,\mathrm{d}S + \frac{1}{\mathrm{R}}\int_{\mathbb{A}\cup\mathbb{I}} \underline{n}\cdot\nabla\underline{u}\,\mathrm{d}S + \int_{\mathbb{V}} \underline{g}\,\mathrm{d}V,$$

represents the integral form of the momentum equation. Here $\underline{g} = (0, 1/\mathrm{Fr}^2, 0)$ and p denotes the reduced pressure p' of equation (1.16). This vector equation has two scalar projections on the X and Y axes. Both projections are discretized using the same techniques. In this subsection the spatial and temporal discretizations of (1.16) are discussed.

Unification of the discretization of vector equation components

An extensive literature survey shows that the discretization of vector equations is usually described for one component only, while the second component is stated to be discretized in the same way (see, for example, Gerrits (2001)). In this thesis, a special notation is introduced to provide a single explanation of the discretization procedure,

which covers both components at the same time. In addition, the discretization of only one component has to be implemented in the computer code. The present code that implements the discretization of the remaining component is generated automatically by the compiler. This is done by introducing the longitudinal (L) and transversal (T) axes. Each component of equation (1.16) is then associated with a mapping that defines the correspondence of the (L,T) axes with the (X,Y) axes. For the X component, the L axis is equivalent to the X axis, and the T axis is equivalent to the Y axis, i.e., $L \equiv X$, $T \equiv Y$. $L \equiv Y$ and $T \equiv X$ for the Y component. The L axis is called the longitudinal axis, since it is always directed along the coordinate that corresponds to the projection of equation (1.16) under consideration. The T axis is called the transversal axis. The velocity components u and v are replaced by u_L and u_T. For the X component, $u_L \equiv u$ and $u_T \equiv v$, while for the Y component, $u_L \equiv v$ and $u_T \equiv u$. Indices i and j that run along the X and Y axes, respectively, are mapped to indices l and t which run along the L and T axes. For the X component, $i \equiv l$ and $j \equiv t$, while for the Y component $i \equiv t$ and $j \equiv l$. Then, the velocity knot $u_{i,j-1}$ is equivalent to $u_{L\,l,t-1}$ for the X component and to $u_{T\,l-1,t}$ for the Y component.

The discretization for the L component of equation (1.16) has the following form

$$\frac{d}{dt}\int_{\mathbb{V}} u_L \,\mathrm{d}V + \int_{\mathbb{A}} u_L\,(\underline{n}\cdot\underline{u})\,\mathrm{d}S + \int_{\mathbb{A}\cup\mathbb{I}} p\,n_L\,\mathrm{d}S - \frac{1}{\mathrm{R}}\int_{\mathbb{A}\cup\mathbb{I}} \underline{n}\cdot\nabla u_L\,\mathrm{d}S - \int_{\mathbb{V}} g_L\,\mathrm{d}V = 0. \quad (2.11)$$

The linear equation that results from the discretization of (2.11) is expressed in (L,T) terms. To obtain the discretization of the X component, one should take $u_L \equiv u$ and $u_T \equiv v$, and transform the (l,t) indices into (i,j) indices. For the Y

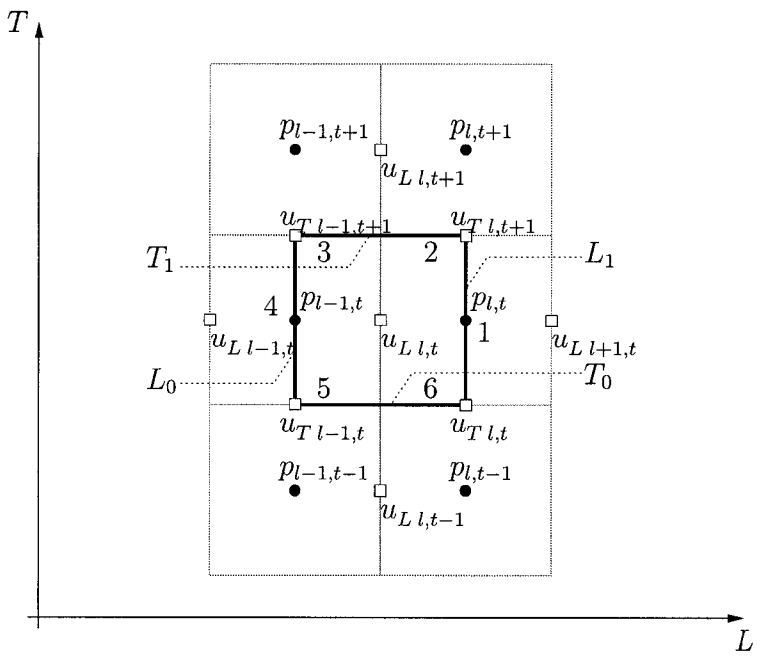

Figure 2.7: Discretization of the momentum equation.

component, $u_L \equiv v$ and $u_T \equiv u$. This procedure seems to be complicated. However, there is no need to obtain discretizations for the X and Y components manually. In implementation, the discretization of both X and Y components is generated by the compiler from the single discretization corresponding to the L component.

Equation (2.11) is discretized for the control volume $U_L(l,t)$ which is shown in Figure 2.7 as a rectangle with bold black edges. The equation consists of unsteady, convection, pressure, diffusion and volumetric force terms. Contributions made by each term are considered separately.

Unsteady term

The unsteady term in (2.11) is

$$\frac{d}{dt}\int_{\mathbb{V}} u_L \ \mathrm{d}V.$$

The following approximations are performed. First, u_L is considered to be constant within volume $\mathbb{V}$. Second, since the velocity knot u_L is located on the edge shared by cells $P(l-1,t)$ and $P(l,t)$, the fractional volume $\mathbb{V}$ corresponding to the cell $U_L(i,j)$ is approximated by the average of the fractional volumes $\mathbb{V}_{l-1,t}$ and $\mathbb{V}_{l,t}$

$$\frac{d}{dt}\int_{\mathbb{V}} u_L \ \mathrm{d}V \approx \frac{u_{L\,l,t}^{k+1}(\mathbb{V}_{l-1,t}^{k+1}+\mathbb{V}_{l,t}^{k+1}) - u_{L\,l,t}^{k}(\mathbb{V}_{l-1,t}^{k}+\mathbb{V}_{l,t}^{k})}{2\Delta t}. \tag{2.12}$$

Convection term

The convective term in (2.11) is

$$\int_{\mathbb{A}} u_L \ (\underline{n}\cdot\underline{u}) \ \mathrm{d}S.$$

This term consists of four fluxes of the L^{th} velocity component through the edges of the control volume $U_L(l,t)$. The edge located in the negative direction of the L axis from the center of the control volume is L_0, whereas the edge located in the positive direction of the L axis is L_1. Two other edges, T_0 and T_1, are defined in a similar way with respect to the T axis. Edges L_0, L_1, T_0 and T_1 corresponding to the control volume $U_L(l,t)$ are shown in Figure 2.7.

For each edge, a characteristic value of u_L as well as a characteristic mass flux F are defined, which leads to the following approximation

$$\int_{\mathbb{A}} u_L \, (\underline{n} \cdot \underline{u}) \, \mathrm{d}S \approx u_L \, F. \tag{2.13}$$

The values of u_L and F are defined for edges L_0, L_1, T_0 and T_1 as follows. For the edge L_0, the characteristic velocity u_L is the average of the L^{th} velocity components located on the edges of the cell $P(l-1,t)$ parallel to the T axis

$$u_L \Big|_{L_0} = \frac{u^{k+1}_{L\,l-1,t} + u^{k+1}_{L\,l,t}}{2}. \tag{2.14}$$

The unweighted average is used due to the reasons explained in the work by Gerrits (2001). The characteristic flux F corresponding to the edge L_0 is the average of the fluxes through the edges L_0 and L_1 of the cell $P(l-1,t)$, which are taken to be negative since the normal $\underline{n}$ corresponding to the edge L_0 of the cell $U_L(l,t)$ is $(-1,0)$

$$F \Big|_{L_0} = -\frac{u^k_{L\,l-1,t}\, \mathbb{A}^k_{L_0\,l-1,t} + u^k_{L\,l,t}\, \mathbb{A}^k_{L_1\,l-1,t}}{2}. \tag{2.15}$$

Here, the velocity components are determined at the time t^k. Thus, their values are known, and the convection term discretization results in a linear equation. The resulting approximation of the convective flux through the edge L_0 is

$$\int_{\mathbb{A}_{L_0}} u_L \, (\underline{n} \cdot \underline{u}) \, \mathrm{d}S \approx -\frac{1}{2}\left(u^{k+1}_{L\,l-1,t} + u^{k+1}_{L\,l,t}\right) \frac{1}{2}\left(u^k_{L\,l-1,t}\, \mathbb{A}^k_{L_0\,l-1,t} + u^k_{L\,l,t}\, \mathbb{A}^k_{L_1\,l-1,t}\right). \tag{2.16}$$

Similarly, the velocity flux through the edge L_1 is approximated as

$$\int_{\mathbb{A}_{L_1}} u_L \, (\underline{n} \cdot \underline{u}) \, \mathrm{d}S \approx \frac{1}{2} \left(u^{k+1}_{L\, l,t} + u^{k+1}_{L\, l+1,t} \right) \frac{1}{2} \left(u^{k}_{L\, l,t} \, \mathbb{A}^{k}_{L_0\, l,t} + u^{k}_{L\, l+1,t} \, \mathbb{A}^{k}_{L_1\, l,t} \right). \tag{2.17}$$

It is noted that approximations (2.16) and (2.17) contain the fractional areas corresponding to the pressure control volumes $P(l-1,t)$ and $P(l,t)$. The fractional areas corresponding to the control volume $U_L(l,t)$ are not used and thus there is no need to compute them.

For the edge T_0, the characteristic velocity u_L is the average of the L^{th} velocity components $u_{L\, l,t-1}$ and $u_{L\, l,t}$

$$u_L \Big|_{T_0} = \frac{u^{k+1}_{L\, l,t-1} + u^{k+1}_{L\, l,t}}{2}. \tag{2.18}$$

The characteristic flux F consists of two terms

$$F \Big|_{T_0} = \frac{1}{2} \mathbb{A}^{k+1}_{T_0\, l-1,t} \, u^{k}_{T\, l-1,t} + \frac{1}{2} \mathbb{A}^{k+1}_{T_0\, l,t} \, u^{k}_{T\, l,t}. \tag{2.19}$$

Thus, velocity flux through the edge T_0 is approximated as

$$\int_{\mathbb{A}_{T_0}} u_L \, (\underline{n} \cdot \underline{u}) \, \mathrm{d}S \approx -\frac{1}{2} (u^{k+1}_{L\, l,t-1} + u^{k+1}_{L\, l,t}) \left(\frac{1}{2} \mathbb{A}^{k+1}_{T_0\, l-1,t} \, u^{k}_{T\, l-1,t} + \frac{1}{2} \mathbb{A}^{k+1}_{T_0\, l,t} \, u^{k}_{T\, l,t} \right). \tag{2.20}$$

Similarly, the velocity flux through the edge T_1 is approximated as

$$\int_{\mathbb{A}_{T_1}} u_L \, (\underline{n} \cdot \underline{u}) \, \mathrm{d}S \approx \frac{1}{2} (u^{k+1}_{L\, l,t+1} + u^{k+1}_{L\, l,t}) \left(\frac{1}{2} \mathbb{A}^{k+1}_{T_1\, l-1,t} \, u^{k}_{T\, l-1,t+1} + \frac{1}{2} \mathbb{A}^{k+1}_{T_1\, l,t} \, u^{k}_{T\, l,t+1} \right). \tag{2.21}$$

Equations (2.16), (2.17), (2.20) and (2.21) are the contributions made by the convective term in (2.11) due to the discretization of the momentum equation.

Diffusion term

The diffusion term in equation (2.11) is

$$\frac{1}{\mathrm{R}} \int_{\text{AUI}} \underline{n} \cdot \nabla u_L \, \mathrm{d}S. \tag{2.22}$$

This term is discretized using a technique introduced by Gerrits (2001), where the integrand is represented in the following form

$$\underline{n} \cdot \nabla u_L = \frac{\partial u_L}{\partial \underline{n}}. \tag{2.23}$$

The normal derivative is approximated as the difference between the characteristic velocity u_L at the boundary of the control volume $U_L(i,j)$ and the velocity component at the control volume center $u_{L\,l,t}$, divided by the distance between these velocities

$$\frac{1}{\mathrm{R}} \int_{\text{AUI}} \frac{\partial u_L}{\partial \underline{n}} \, \mathrm{d}S \approx \frac{1}{\mathrm{R}} \int_{\text{AUI}} \frac{u_L - u_{L\,l,t}}{|\underline{n}|} \, \mathrm{d}S. \tag{2.24}$$

The control volume boundary is split into eight parts: segment 1 corresponds to the edge L_1, segments 2 and 3 correspond to the edge T_1, segment 4 corresponds to the edge L_0, segments 5 and 6 correspond to the edge T_0, and segments 7 and 8 correspond to the fluid-solid interfaces located in cells $P(l-1,t)$ and $P(l,t)$, respectively (if there is no fluid-solid interface, its length is set to zero, which results in no contribution

made by the interface terms). Segments 1 - 6 are shown in figure 2.7. For each segment k, the distance $|\underline{n}_k|$ which has the dimension of the length is approximated as a certain volume V_k corresponding to the segment divided by the length A_k of the segment

$$|\underline{n}_k| = \frac{V_k}{A_k}. \tag{2.25}$$

Substitution of (2.25) into (2.24) for segment k gives

$$\frac{1}{\mathrm{R}} \int_k \frac{\partial u_k}{\partial \underline{n}} \, \mathrm{d}S \approx \frac{1}{\mathrm{R}} (u_k - u_{L\,l,t}^{k+1}) \frac{A_k^2}{V_k}. \tag{2.26}$$

Here, u_k is the L^{th} velocity component at segment k. The central velocity $u_{L\,l,t}$ is taken at time t^{k+1}. Once u_k, A_k and V_k corresponding to all segments are defined, their substitution into (2.26) will give the contributions made by the diffusion term. u_k, A_k and V_k are chosen according to Gerrits's work as follows

$$u_1 = u_{L\,l+1,t}^{k+1}, \quad A_1 = \frac{1}{2}(\mathbb{A}_{L_0\,l,t}^{k+1} + \mathbb{A}_{L_1\,l,t}^{k+1}), \quad V_1 = \mathbb{V}_{l,t}^{k+1}; \tag{2.27a}$$

$$u_2 = u_{L\,l,t+1}^{k+1}, \quad A_2 = \frac{1}{2}\mathbb{A}_{T_1\,l,t}^{k+1}, \quad V_2 = \frac{1}{4}(\mathbb{V}_{l,t}^{k+1} + \mathbb{V}_{l,t+1}^{k+1}); \tag{2.27b}$$

$$u_3 = u_{L\,l,t+1}^{k+1}, \quad A_3 = \frac{1}{2}\mathbb{A}_{T_1\,l-1,t}^{k+1}, \quad V_3 = \frac{1}{4}(\mathbb{V}_{l-1,t}^{k+1} + \mathbb{V}_{l-1,t+1}^{k+1}); \tag{2.27c}$$

$$u_4 = u_{L\,l-1,t}^{k+1}, \quad A_4 = \frac{1}{2}(\mathbb{A}_{L_0\,l-1,t}^{k+1} + \mathbb{A}_{L_1\,l-1,t}^{k+1}), \quad V_4 = \mathbb{V}_{l-1,t}^{k+1}; \tag{2.27d}$$

$$u_5 = u_{L\,l,t-1}^{k+1}, \quad A_5 = \frac{1}{2}\mathbb{A}_{T_0\,l-1,t}^{k+1}, \quad V_5 = \frac{1}{4}(\mathbb{V}_{l-1,t}^{k+1} + \mathbb{V}_{l-1,t-1}^{k+1}); \tag{2.27e}$$

$$u_6 = u_{L\,l,t-1}^{k+1}, \quad A_6 = \frac{1}{2}\mathbb{A}_{T_0\,l,t}^{k+1}, \quad V_6 = \frac{1}{4}(\mathbb{V}_{l,t}^{k+1} + \mathbb{V}_{l,t-1}^{k+1}); \tag{2.27f}$$

$$u_7 = u_{L\,\text{body}}^{k+1}, \quad A_7 = \frac{1}{2}\mathbb{I}_{l-1,t}^{k+1}, \quad V_7 = \frac{1}{2}\mathbb{V}_{l-1,t}^{k+1}; \tag{2.27g}$$

$$u_8 = u_{L\,\text{body}}^{k+1}, \quad A_8 = \frac{1}{2}\mathbb{I}_{l,t}^{k+1}, \quad V_8 = \frac{1}{2}\mathbb{V}_{l,t}^{k+1}. \tag{2.27h}$$

In (2.27g) and (2.27h), $u_{L\,\text{body}}^{k+1}$ is the body velocity at the midpoint of the interface segments $\mathbb{I}_{l-1,t}^{k+1}$ and $\mathbb{I}_{l,t}^{k+1}$ located in the cells $P(l-t,t)$ and $P(l,t)$, respectively.

Although (2.27a) - (2.27h) are obtained based on the work by Gerrits (2001), there are three minor differences due to different definitions of the geometry of the control volumes partly occupied by the solid body. First, A_1 and A_4 in (2.27a) are computed by averaging, while in Gerrits' work these quantities are computed as the areas of control volume boundary located in the cells $P(l,t)$ and $P(l-1,t)$, respectively. Second, u_7 and u_8 in equations (2.27g) and (2.27h) are computed using the solid body velocity, whereas a mirrored velocity is used in Gerrits' work based on the assumption that body velocity is always zero. Third, in this work interface lengths A_7 and A_8 are approximated with half of the interfaces passing through the cells $P(l-1,t)$ and $P(l,t)$, while in Gerrits' work each of these interfaces is considered as a whole or not considered at all, depending on control volume geometry.

Substitution of (2.27a) - (2.27h) into (2.26) gives the contributions made by the diffusion term in (2.11) due to the discretization of the momentum equation.

Pressure term

The pressure term in (2.11) is

$$\int_{\mathbb{A}\cup\mathbb{I}} p\, n_L \,\mathrm{d}S. \tag{2.28}$$

Since only the L^{th} component of the normal to the boundary appears in this equation, integration over the edges T_0 and T_1 gives zero. Following the work of Gerrits (2001),

the parts of the boundary in the cells $P(l-1,t)$ and $P(l,t)$ not parallel to the L axis are both approximated as $\mathbb{A}^{k+1}_{L_0\,l,t}$, which results in the following approximation of the pressure term contribution

$$\int_{\mathbb{A}\cup\mathbb{I}} p\, n_L \,\mathrm{d}S \approx (p^{k+1}_{l,t} - p^{k+1}_{l-1,t})\mathbb{A}^{k+1}_{L_0\,l,t}. \tag{2.29}$$

Volumetric force term

The volumetric force term in (2.11) is

$$\int_{\mathbb{V}} g_L \,\mathrm{d}V. \tag{2.30}$$

As it is pointed out in the work by Gerrits (2001), the discretization of the volumetric force term should be consistent with the discretization of the pressure term. This is due to the fact that a static fluid in the presence of gravity force is considered and thus, the discretized volumetric force cancel the hydrostatic pressure term. This affects the computation of the volume $\mathbb{V}$ in equation (2.30). Taking the length parallel to the T axis as $\mathbb{A}^{k+1}_{L_0\,l,t}$ gives

$$\int_{\mathbb{V}} g_L \,\mathrm{d}V \approx g_L \mathbb{A}^{k+1}_{L_0\,l,t}(X^0_l - X^0_{l-1}). \tag{2.31}$$

Here, X^0_l is the L^{th} coordinate of the pressure knot whose L-index is equal to l, so $(X^0_l - X^0_{l-1})$ is the length of the control volume along the L axis.

2.5 Boundary conditions at fluid interfaces

The discretization of the governing equations (1.10) and (2.11) is done only inside the fluid region of the computational domain. At the free surface and at the solid body interface, the boundary conditions, presented in Section 1.3, are discretized.

2.5.1 Pressure interpolation at the free surface

In S-cells, a value for the pressure is needed. This is provided by boundary condition (1.19) for the pressure at the free surface

$$\frac{2}{\mathrm{R}}\frac{\partial u_n}{\partial \underline{n}} - p = -p_0.$$

The first term $\left(\frac{2}{\mathrm{R}}\frac{\partial u_n}{\partial \underline{n}}\right)$ on the left hand side of this equation is neglected due to the relatively small viscosity assumption used in this thesis, and the value of the normal velocity derivative is typically very small (see, for example, Hirt and Nichols (1981), Gerrits (2001)). Thus, the value of the pressure at the free surface, p, is equal to the ambient air pressure, p_0, i.e.,

$$p = p_0. \tag{2.32}$$

This equation is discretized in every S-cell by defining the pressure in this type of cell. This is done based on the interpolation of the pressure at the free surface and the pressure in the neighbouring fluid cell, p_N. This method is illustrated in Figure 2.8, where S-cell $P(i,j)$ under consideration is referred to as the base cell, p_B. The free surface orientation is approximated in the vicinity of the base cell by computing the

following differences of the VOF fractions along the X and Y axes

$$\Delta x = |F^{k+1}_{i+1,j} - F^{k+1}_{i-1,j}|, \qquad \Delta y = |F^{k+1}_{i,j+1} - F^{k+1}_{i,j-1}|. \tag{2.33}$$

The orientation of the free surface is stated to be horizontal if $\Delta x < \Delta y$, or vertical, otherwise. The neighbouring pressure cell of type F is selected based on the free surface orientation. This neighbouring cell, p_N, is also referred to as *the interpolation neighbour of the surface cell.* In the case of the horizontal free surface, cell p_N is either $P(i-1,j)$ or $P(i+1,j)$, whereas for the vertical free surface orientation p_N is either $P(i,j-1)$ or $P(i,j+1)$. In Figure 2.8 the horizontal free surface orientation is chosen and cell p_N is $P(i,j-1)$. Since the free surface is perpendicular to the line

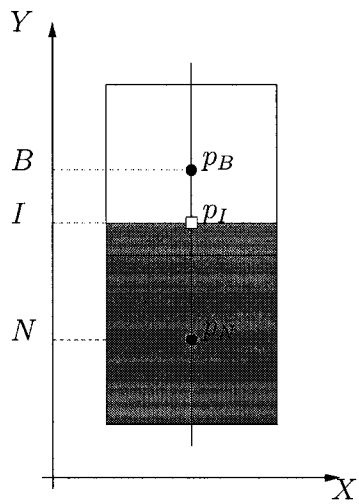

Figure 2.8: Interpolation of the free surface pressure, p_I, to the S pressure knot, p_B, using the neighbouring fluid pressure, p_N.

that joins pressure knots p_N and p_B, the interface location p_I along this line can be computed from the known fluid fractions in the base cell and neighbouring cell (see Figure 2.8). Once the coordinates B, N and I are known, the pressure p_B in the base

cell $P(i,j)$ can be computed based on linear interpolation of p_I and p_N

$$(I-N)p_B + (B-I)p_N + (N-B)p_I = 0. \tag{2.34}$$

The interfacial pressure, p_I, is equal to the ambient air pressure p_0. Thus, the discretization of (1.19) yields the linear equation for $p_{i,j}^{k+1}$

$$(I-N)p_{i,j}^{k+1} + (B-I)p_N^{k+1} + (N-B)p_0 = 0. \tag{2.35}$$

2.5.2 Extension of the velocity field to the free surface

The velocities of type S can appear in the discretization of the momentum equation (1.16) corresponding to the neighbouring velocity knots and thus require special consideration. In this thesis, S-velocities are computed following the work by Gerrits (2001). The S-velocity knot $u_{L\,l,t}$ is considered in Figure 2.9. According to the cell and knot classification described in Section 2.3, one of the pressure cells $P(l-1,t)$ or $P(l,t)$ is a S-cell as shown in Figure 2.9. If $u_{L\,l,t}$ is the only velocity of type S in the S-cell, this velocity can be obtained from the mass conservation discretization pattern applied to this S-cell. Otherwise, $u_{L\,l,t}$ can be set equal to the neighbouring fluid velocity knot $u_{L\,\mathrm{nbr}}$ in the fluid direction

$$u_{L\,l,t}^{k+1} - u_{L\,\mathrm{nbr}}^{k+1} = 0, \tag{2.36}$$

where $u_{L\,\mathrm{nbr}}^{k+1}$ is one of $u_{L\,l-1,t}^{k+1}$, $u_{L\,l+1,t}^{k+1}$, $u_{L\,l,t-1}^{k+1}$ and $u_{L\,l,t+1}^{k+1}$ as shown in Figure 2.9.

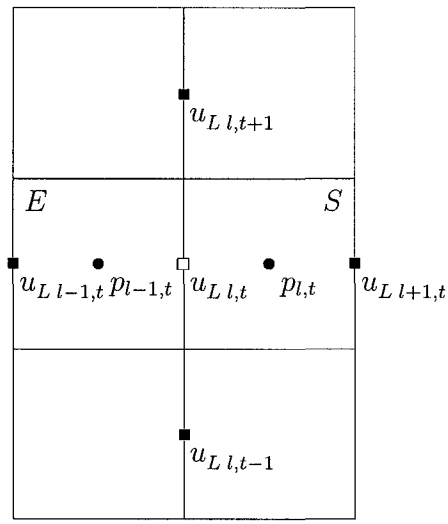

Figure 2.9: S-velocity discretization. In the case when conservation of mass cannot be used, $u_{L\,l,t}$ is set equal to the neighbouring fluid knot chosen from the knots marked as black squares.

2.5.3 Tangential velocity boundary condition at the free surface

The majority of the velocity knots of type E are located in the air and thus are set to zero: $u_{L\,l,t}^{k+1} = 0$. However, certain E-velocities are required in the discretizations of the momentum equation for the neighbouring S-velocity knots. An example is shown in Figure 2.10, where E-velocity $u_{L\,l,t}$ is required in the discretization pattern corresponding to S-velocity $u_{L\,l,t-1}$. Following the work by Gerrits (2001), the tangential boundary condition (1.18) is used to define such E-velocities in this thesis. The free surface is assumed to be horizontal if E-velocity is the X-velocity component, and vertical if E-velocity is the Y-velocity component. Thus, the normal and tangential directions are always aligned with the coordinate axes, and equation (1.18) can be

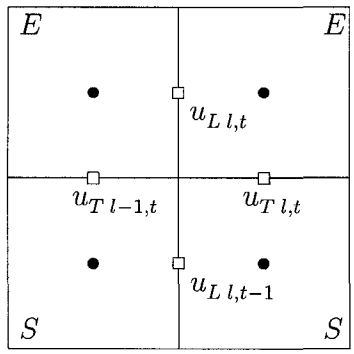

Figure 2.10: Application of the tangential boundary condition for velocity knot $u_{L\,l,t}$.

discretized using finite differences.

The case when the free surface is horizontal is illustrated in Figure 2.10 so that $\frac{\partial}{\partial \underline{n}} = \frac{\partial}{\partial t}$, $\frac{\partial}{\partial \underline{\tau}} = \frac{\partial}{\partial l}$, and the discretization results in

$$\frac{u^{k+1}_{T\,l,t} - u^{k+1}_{T\,l-1,t}}{x^T_l - x^T_{l-1}} + \frac{u^{k+1}_{L\,l,t} - u^{k+1}_{L\,l-1,t}}{x^L_t - x^L_{t-1}} = 0. \tag{2.37}$$

Here, x^T_l is the L^{th} coordinate of the knot $u_{T\,l,t}$, and x^L_t is the T^{th} coordinate of the knot $u_{L\,l,t}$.

2.5.4 No-slip boundary condition

The no-slip boundary condition given by equation (1.17) states that the fluid velocity at each point of the fluid-solid interface should be equal to the solid velocity at that point. Due to the use of a non-boundary fitted grid in this work, the velocity knots are not located at the fluid-solid interface. This complicates the discretization of the no-slip boundary condition. In this work, a method similar to the one presented in

Section 2.5.1 for the pressure interpolation at the free surface is used. The technique for discretizing the no-slip boundary condition for the velocity knot $u_{L\,l,t}$ is illustrated in Figure 2.11. The knot $u_{L\,l,t}$ is labeled as the base knot u_B. Four neighbours $u_{L\,l-1,t}$, $u_{L\,l+1,t}$, $u_{L\,l,t-1}$ and $u_{L\,l,t+1}$ are searched for a neighbouring fluid knot (i.e., for the velocity knot of type F). If no fluid neighbour is found, then the velocity $u_{L\,l,t}$ is set to the solid body velocity. The neighbouring fluid knot is denoted as u_N. The fluid-solid interface is approximated as the perpendicular to the line that joins knots u_B and u_N as shown in Figure 2.11. The line that joins u_B and u_N is either parallel

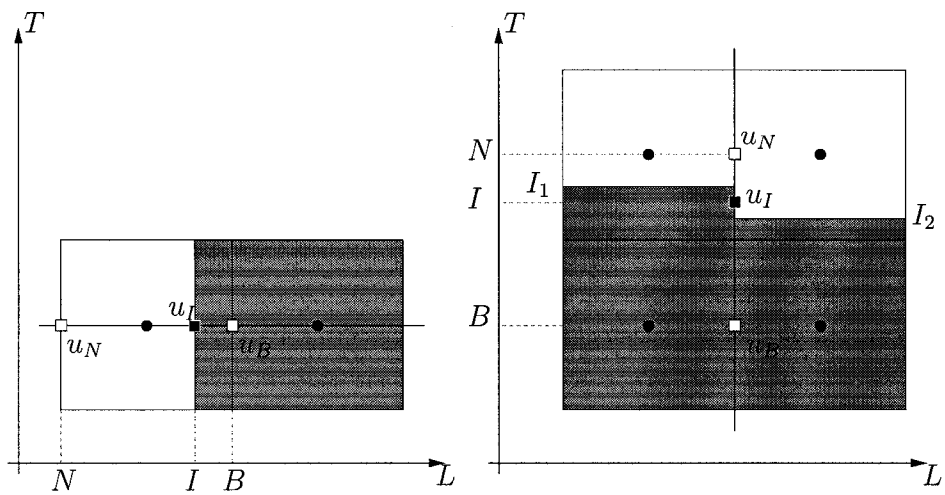

Figure 2.11: No-slip boundary condition: u_B is the base velocity knot, u_N is the neighbour fluid velocity knot, I is the estimation of location of the fluid-solid interface. Approximated solid fractions are shown with gray color.

or perpendicular to the L axis as displayed in Figure 2.11. In the case when this line is parallel to the L axis, the coordinate of the approximate interface location I in the direction of the line is obtained from the known solid fractions in the pressure cells that have the base velocity, u_B, on one of their edges. In the latter case, I is obtained as the mean value of two estimated interface locations I_1 and I_2.

Once the coordinates B, N and I along the axis parallel to the line that joins u_B and u_N are determined, the velocity u_B at the base knot can be computed from linear interpolation of $u_N = u^{k+1}_{L\,N}$ and $u_I = u^{k+1}_{L\,solid}$

$$(N-I)u^{k+1}_{L\,B} + (I-B)u^{k+1}_{L\,N} + (B-N)u^{k+1}_{L\,solid} = 0, \tag{2.38}$$

where $u^{k+1}_{L\,solid}$ is the L^{th} component of the solid velocity at the point I. All the velocities are taken at time t^{k+1}. This interpolation may lead to unphysical high values of the base velocity when the fluid-solid interface I is close to the neighbour velocity knot N. To prevent this, (2.38) is applied only when $|B-I| < |N-I|$. Otherwise, the base velocity is set to the solid body velocity.

2.6 Boundary conditions at the computational domain boundary

The numerical method presented in this Chapter aims to solve the two-dimensional, unsteady, incompressible Navier-Stokes equations in their integral form in a semi-infinite domain to model the hydrodynamic environment around a transversely oscillating cylinder in the presence of a free surface. The numerical solution of unbounded problems requires the proper representation of the boundary conditions after representing the unbounded domain by a finite computational domain.

The computational domain boundary, shown in Figure 2.12, is divided into four parts: the inflow boundary on the left; the outflow boundary on the right; the inviscid wall

boundary located at the bottom which truncates the infinite fluid region at a certain depth and the inviscid wall boundary at the top which truncates the upper part of the computational domain. Both the lower and upper inviscid wall boundaries

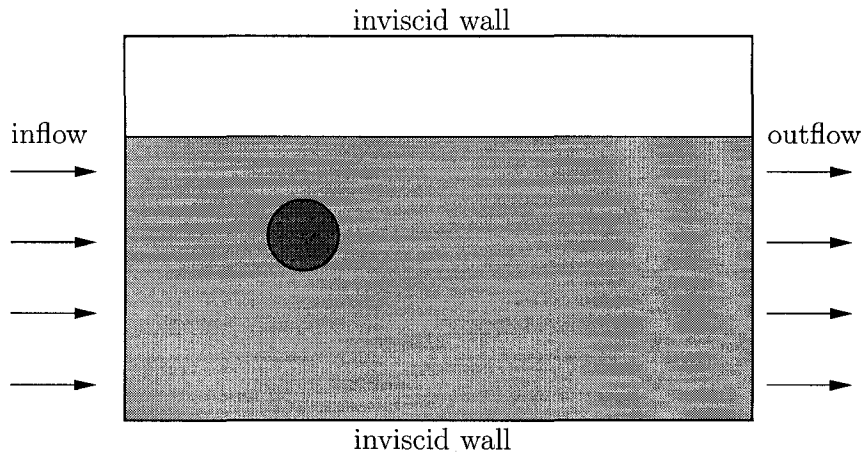

Figure 2.12: The computational domain boundaries.

are modeled similarly to enable numerical simulations of a transversely oscillating cylinder in this infinite medium of fluid.

The velocity knots of type D are located on the edges of D-cells. D-velocity knots which are located on the computational domain boundary or belong to the corner cells in the computational domain are not used in other discretization patterns and thus can be set to zero. These excluded knots are shown in Figure 2.13. All the remaining velocity knots of type D are used to apply the boundary conditions on the computational domain boundary. In what follows, the boundary conditions are described and discretized for each of the four walls of the domain boundary.

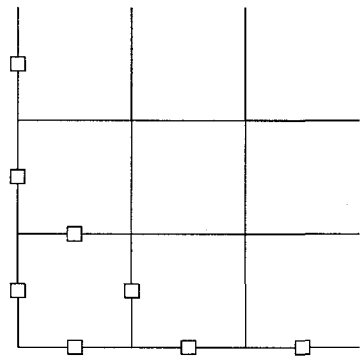

Figure 2.13: Excluded knots on the computational domain boundary.

Inflow boundary condition

At the inflow boundary, the fluid velocity is denoted as

$$\underline{u} = \underline{u}_0, \tag{2.39}$$

where $\underline{u}_0$ is a prescribed velocity. It is noted that when the inflow boundary is placed far away from the cylinder, this boundary condition provides a good approximation of the mathematical boundary condition $\underline{u} = \underline{u}_0$ at infinity. Thus, at this wall, both

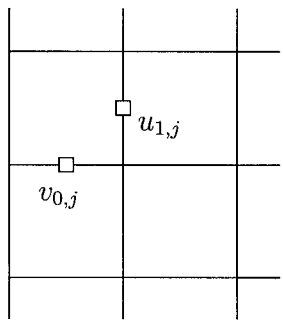

Figure 2.14: The velocity knots used to apply the inflow boundary condition.

components of the fluid velocity are set equal to the prescribed velocity, as shown in

Figure 2.14,

$$u_{1,j}^{k+1} = u_0, \qquad v_{0,j}^{k+1} = v_0. \tag{2.40}$$

Inviscid wall boundary condition

At a sufficiently large depth below the free surface, the fluid flow is approximated by the uniform flow, i.e., streamlines are represented by straight lines. Thus, one of these streamlines is chosen as the computational domain boundary, which acts as an inviscid wall, and at this wall the tangential velocity v does not change, i.e., $\dfrac{\partial v}{\partial y} = 0$. Thus, only the normal fluid velocity u is prescribed at this wall. Figure 2.15 displays

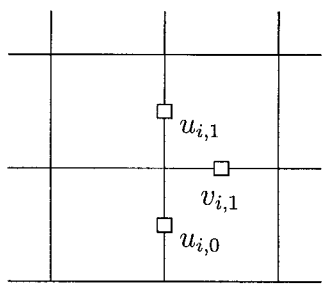

Figure 2.15: The velocity knots used in the inviscid wall boundary condition at the bottom boundary of the computational domain.

the velocity knots involved at the bottom wall, and their discretization is given by

$$u_{i,1}^{k+1} - u_{i,0}^{k+1} = 0, \qquad v_{i,1}^{k+1} = v_0. \tag{2.41}$$

Similarly, the discretization for the inviscid wall boundary condition at the top wall is given by

$$u_{i,M-1}^{k+1} - u_{i,M-2}^{k+1} = 0, \qquad v_{i,M-1}^{k+1} = v_0. \tag{2.42}$$

Here, M is the number of pressure cells along the Y axis.

Outflow boundary condition

The outflow boundary condition is applied at the right boundary of the computational domain as shown in Figure 2.16. The normal and tangential velocity components,

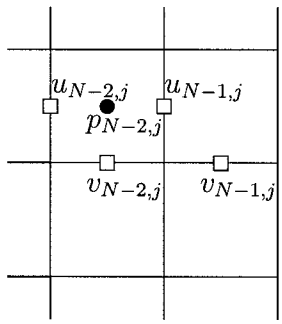

Figure 2.16: The pressure and velocity knots used in the application of the outflow boundary condition.

$u_{N-1,j}$ and $v_{N-1,j}$, correspond to boundary conditions

$$\int_S \left(\frac{1}{\mathrm{R}} \frac{\partial u}{\partial x} - p \right) \mathrm{d}S = - \int_S \left(p_0 + \frac{h}{\mathrm{Fr}^2} \right) \mathrm{d}S \tag{2.43}$$

and

$$\int_S \frac{\partial v}{\partial x} \mathrm{d}S = 0, \tag{2.44}$$

respectively. Derivation of these boundary conditions will be reported elsewhere. They are approximated based on the assumption that the integrands in 2.43 and 2.44

are constant along the boundary S. This results in discretized equations

$$\frac{1}{\mathrm{R}}\frac{u^{k+1}_{N-1,j}-u^{k+1}_{N-2,j}}{x^1_{N-1}-x^1_{N-2}}-p^{k+1}_{N-2,j}=-p_0-\frac{h}{\mathrm{Fr}^2},\qquad v^{k+1}_{N-1,j}-v^{k+1}_{N-2,j}=0. \tag{2.45}$$

Here, the depth of submergence, h, can be computed from either the actual or the required fluid elevation at the outflow at time t^{k+1}.

2.7 Initial conditions

The cylinder motion starts impulsively from the rest at $t = 0$. The potential flow assumption is made, i.e,

$$u(x,y)=U\left(1-\frac{d^2(x^2-y^2)}{4(x^2+y^2)^2}\right),\qquad v(x,y)\ =\frac{-Uxyd^2}{2\,(x^2+y^2)^2} \tag{2.46}$$

at the initial time $t = 0$. At time $t = 0^+$, the free-slip velocity boundary condition at the cylinder surface, suddenly changes to the no-slip condition, $\underline{u} = \underline{u}_{\mathrm{body}}$, due to the viscous flow development. This results in development of boundary layers at the cylinder surface, where the potential flow velocity rapidly decays to zero. As it is well known, the initial flow is governed by the usual boundary-layer theory in which a layer of thickness $(t/\mathrm{R})^{1/2}$ surrounds the cylinder following the sudden start at $t = 0$.

2.8 Correspondence between cell classification and discretization patterns

The types of pressure cells and velocity knots and their discretization patterns are displayed in Table 2.2. Subsection numbers corresponding to non-trivial discretization patterns are provided in parentheses.

Pressure cell type	Discretization pattern
E	$p_{i,j}^{k+1} = 0$
F	conservation of mass equation: Section (2.4.2)
X	$p_{i,j}^{k+1} = 0$
S	pressure interpolation at the free surface: Section (2.5.1)
B	$p_{i,j}^{k+1} = 0$
D	$p_{i,j}^{k+1} = 0$
Velocity knot type	**Discretization pattern**
E	tangential velocity boundary condition at the free surface: Section (2.5.3)
F	the momentum equation: Section (2.4.3)
X	$u_{L\,l,t}^{k+1} = 0$
S	extension of the velocity field to the free surface: Section (2.5.2)
B	no-slip boundary condition: Section (2.5.4)
D	velocity boundary conditions at the computational domain boundary: Section (2.6)

Table 2.2: Correspondence between the pressure cell/velocity knot types and discretization patterns.

2.9 Simulation workflow

The numerical simulation begins with the initialization of fluid velocity components $u^0_{i,j}$, $v^0_{i,j}$ and the values of the VOF function $F^0_{i,j}$ at time t^0, which are taken either from the known initial conditions or from previously computed results. It is noted that the discrete pressure field $p^0_{i,j}$ does not need to be initialized since there are no explicit pressure contributions in the discretization of the governing equations and boundary conditions. The values of the VOF function $F^0_{i,j}$ are used to reconstruct the fluid interface and compute the fractional areas $\mathbb{A}^0_{i,j}$ and volumes $\mathbb{V}^0_{i,j}$ corresponding to all computational cells. The pressure cells and the velocity knots are then assigned labels $\mathbb{L}^0_{p,i,j}$, $\mathbb{L}^0_{u,i,j}$, $\mathbb{L}^0_{v,i,j}$ according to the procedure described in Section 2.3.

After this initialization procedure, the computation of the discrete pressure and velocity fields $p^{k+1}_{i,j}$, $u^{k+1}_{i,j}$, $v^{k+1}_{i,j}$ as well as the VOF fractions $F^{k+1}_{i,j}$ corresponding to successive time instants t^k, $k = 0 \ldots K$ is performed. The computation is divided into cycles. On each cycle, the computations required to advance the solution to the next time instant are done. Computational steps corresponding to one cycle are displayed in Figure 2.17. The first step consists of the advection of the VOF function $F^k_{i,j}$ with local fluid velocity $(u^k_{i,j}, v^k_{i,j}, 0)$. As a result of this, the values of the VOF function $F^{k+1}_{i,j}$ corresponding to the new time instant, t^{k+1}, are obtained. Then, the new values $F^{k+1}_{i,j}$ are used to reconstruct the fluid interface and compute fractional areas $\mathbb{A}^{k+1}_{i,j}$ and volumes $\mathbb{V}^{k+1}_{i,j}$. The third step consists of assigning the new labels $\mathbb{L}^{k+1}_{p,i,j}$, $\mathbb{L}^{k+1}_{u,i,j}$, $\mathbb{L}^{k+1}_{v,i,j}$ to the pressure cells and the velocity knots based on the values $F^{k+1}_{i,j}$. In the fourth step a linear system is formed by applying discretization patterns corresponding to every pressure and velocity knot. The linear system is then solved which

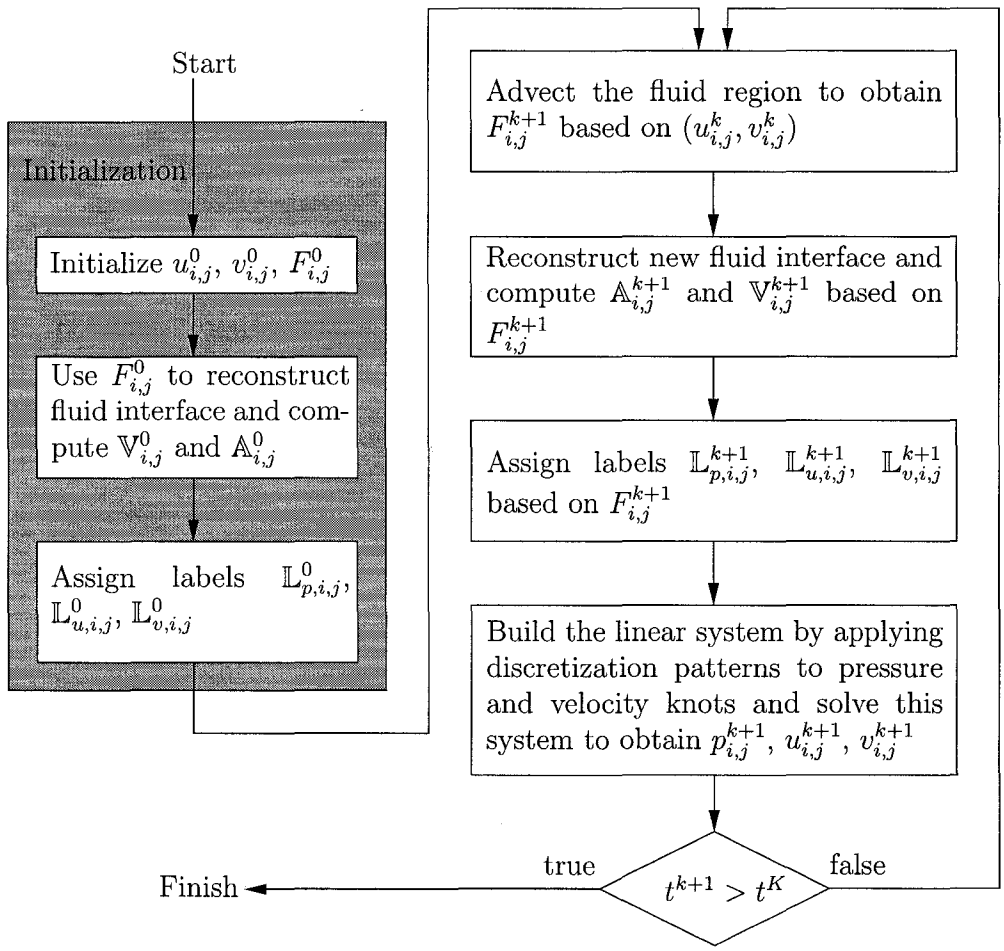

Figure 2.17: Simulation workflow.

results in the discrete pressure and velocity fields $p^{k+1}_{i,j}$, $u^{k+1}_{i,j}$, $v^{k+1}_{i,j}$ corresponding to the new time level t^{k+1}. After the determination of the pressure and velocity, the new time t^{k+1} is checked against the final time t^K to determine whether to continue or stop the numerical simulation. Before entering the next computational cycle, all the variables $p^{k+1}_{i,j}$, $u^{k+1}_{i,j}$, $v^{k+1}_{i,j}$, $F^{k+1}_{i,j}$, $\mathbb{L}^{k+1}_{p,i,j}$, $\mathbb{L}^{k+1}_{u,i,j}$, $\mathbb{L}^{k+1}_{v,i,j}$ are renamed to $p^k_{i,j}$, $u^k_{i,j}$, $v^k_{i,j}$, $F^k_{i,j}$, $\mathbb{L}^k_{p,i,j}$, $\mathbb{L}^k_{u,i,j}$, $\mathbb{L}^k_{v,i,j}$, respectively.

Chapter 3

CFD code design: Implementation and validation

The development of a powerful and efficient high-level framework for simulations of Navier-Stokes problems is an active research topic. In this context, to be effective we explore the implementation of principle CFD concepts into a computer program to solve the free surface problem under consideration. This is done by making use of the numerical techniques described in the previous Chapter and a high-level programming language `C++`. There are two reasons for choosing `C++` as the programming language: its performance and expressiveness. *Performance* plays the greatest role in computational fluid dynamics since results are often limited by available computing resources. A good use of the language *expressiveness* results in a computer program that is easier to understand and maintain. `C++` provides high-level abstraction mechanisms while retaining and even exceeding the performance of the `Fortran` programming language

(see, for example, Stroustrup (2004)). The use of a high-level programming language C++ allows us to represent concepts in the application domain of computational fluid dynamics such as computational grids, discretization patterns and linear equations as C++ classes. Classes are used to encapsulate object behaviour and make it available to other objects by means of a well-defined interface. This results in a computer program which is easier to comprehend and maintain than a program expressed using low-level concepts such as arrays, data structures and simple loops. Powerful features of C++ templates as well as modern compilers that provide optimizations specific to C++ can be employed to achieve the performance comparable with the performance of the Fortran programming language. The C++ classes and templates do not introduce performance penalty, unlike other object-oriented languages such as Java and Python. For example, Java templates are run-time entities, while C++ templates are solely used at compile-time. An expensive feature such as *run-time type information* is optional in C++, while it is embedded in Java and Python.

The code written for this work employs a combination of various programming styles supported by C++. The *data abstraction* paradigm is used to create a data type suitable for storing the values of the computed pressure and velocity. The use of a *generic programming* paradigm facilitates the implementation of discretization patterns and reduces the size of the corresponding code up to four times. During compile time, the code corresponding to both components of the vector equations is generated from a single template by the C++ compiler. An *object-oriented* paradigm is applied to represent the code as a set of classes, where each class is responsible for a distinct task. During run time, objects of these classes are created to perform a fluid flow simulation in collaboration with each other. In the subsequent sections of this Chapter, the use

of these paradigms is described in detail.

3.1 Data type for pressure and velocity variables

The values of the pressure $p_{i,j}^{k+1}$ and velocity components $u_{i,j}^{k+1}$, $v_{i,j}^{k+1}$ are computed at each time step as the solution of a system of algebraic equations of the form

$$Ax = b. \tag{3.1}$$

Here, $A = (a_{i,j})$ is the matrix of this linear system, b is the vector of free terms, and x is the vector of unknowns. Since unknowns are stored in a one-dimensional vector x, a single index is associated with each pressure and velocity component $p_{i,j}^{k+1}$; $u_{i,j}^{k+1}$, $v_{i,j}^{k+1}$. This ordering of unknowns influences the sparsity pattern of A, which, in turn, may affect the time and memory required to solve the system of equations (3.1) (see, for example, Benzi (2002)). The values of the pressure and velocity components computed as the solution of (3.1) are used to form the system of equations for the next time step. At this time, it is more convenient to refer to these values using multiple indices rather than a single one. Thus, the vector of variables possesses a dual behaviour, i.e., it is a one-dimensional array and a set of two-dimensional matrices at the same time.

In this thesis, the `PolyMatrixArray` abstract data type suitable for storing and accessing the vector of unknowns is introduced. Stroustrup (2000) showed that `std::valarray` can be used as a container for all variables which is optimized for numerical cal-

culations. An effective way of addressing within this container is provided by the `std::slice` class. Using these as building blocks, the `PolyMatrixArray` class is created which can be seen as a one-dimensional array and a set of two-dimensional matrices at the same time

```
enum{P = 0, U, V};

void a_function(PolyMatrixArray & a)
{
    a[P][3][2] = 7.0; //use matrix notation
    size_t i = a.index(P, 3, 2);
    a.array()[i] = 14.0; //use vector notation
}
```

Here, the value of $p_{3,2}$ is set to 7.0. The single index corresponding to this knot is then computed, and the `PolyMatrixArray` object is finally accessed as a one-dimensional array to set $p_{3,2}$ equal to 14.0. In addition to this easy access provision to pressure and velocity variables, `PolyMatrixArray` encapsulates a particular ordering of unknowns. In the case where there is a need to change the ordering, modifications are done within this class only so that the rest of the code is not affected. Triple index addressing is implemented as follows

```
class PolyMatrixArray
{
    std::valarray<double> array;
public:
    typedef PolySliceIterator iterator;
    iterator operator[](size_t i);
    std::valarray<double> & array();
};

class PolySliceIterator
{
public:
    typedef SliceIterator iterator;
    iterator operator[](size_t i);
};

class SliceIterator
{
public:
    double & operator[](size_t i);
```

```
};
```

Expression `a[P][i][j]` is parsed as `((a[P])[i])[j]`. Subexpression `a[P]` returns a `PolySliceIterator` object, which represents the `a`-th matrix within `a`. If this object is denoted as `psi`, then our expression becomes `(psi[i])[j]`. Object `psi` has its own `operator[]`, which results in another object `si` of type `SliceIterator`. The expression turns out to be `si[j]`. Object si has its own `operator[]`, which finally returns the reference to element `a[P][i][j]`. In the first instance, this procedure may not seem to be efficient. However, this is not the case. First, all `operator[]` methods are inlined. The *name return value optimization* is then used to prevent construction of copies of iterator objects returned by these methods. Iterator objects are created on the stack, so no dynamic memory allocation takes place. As a result, the tests show no difference in performance between indexing with a single index and indexing with a triple index.

The `PolyMatrixArray` class and its iterators do optional range checks. When compiled in debug mode, each `operator[]` method verifies that the corresponding index is not out of bounds. The standard `assert` macro is used for verification. In release mode, this macro expands to nothing, so no run time penalty appears in this case.

The iterators described above can be viewed as lightweight objects that represent certain parts of the entire array

```
void f(PolyMatrixArray & a)
{
    PolyMatrixArray::iterator P(a[0]);
    P[3][2] = 4.0;
}
```

In function `f()` only the pressure variable is used, so it is convenient to define an object which refers to the pressure submatrix of the entire array. This submatrix is then used as a two-dimensional matrix. One can even pass `P` to another function by value, which is equivalent to the creation of another reference to unknown variables stored in a single location in memory. This is both efficient and convenient.

3.2 Discretized equations

Discretization of the governing equations and boundary conditions results in linear equations in unknowns $p_{i,j}^{k+1}$; $u_{i,j}^{k+1}$, $v_{i,j}^{k+1}$. These equations are obtained by considering the contributions made by various terms such as unsteady, convective, diffusive and volumetric force terms. An object of class `DiscretizationEquation` is used to accumulate these contributions in the computer code

```
void discretize(DiscretizationEquation * eq, size_t i, size_t j)
{
    //start with trivial linear equation "0 = 0"
    eq->reset();

    //add contributions
    eq->add_explicit( 1.0, P, i, j);
    eq->add_implicit(-1.0, P, i, j);
    eq->add_free_term(-2.0);
}
```

Here, three contributions are added: $1 \cdot p_{i,j}^{k}$, $-1 \cdot p_{i,j}^{k+1}$ and -2. As a result, the constant term becomes $p_{i,j}^{k} - 2$ and -1 appears at the column corresponding to $p_{i,j}$. It is noted that one can reset the `DiscretizationEquation` object before putting entire new equation into it. This enables reuse of a single `eq` object throughout the entire simulation.

After all the terms are processed and all contributions are taken into consideration, the `DiscretizationEquation` object is used to fill up the current row in the sparse matrix A and the free term vector b in the linear system (3.1). Specifically, for each implicit variable in the `DiscretizationEquation` object, the single index is computed. Then, the coefficient corresponding to the variable is inserted into the current row of the matrix A. The column number of this coefficient is taken to be the single index of the variable. The free term in the `DiscretizationEquation` object is inserted into the current row of the vector b. The equation corresponding to the next unknown is then discretized, and the resulting `DiscretizationEquation` object is used to fill the next row. Thus, the `DiscretizationEquation` object decouples the part of the code which provides discretization routines and the part of the code which assembles and solves the system of linear equations (3.1).

3.3 Explicit and implicit time levels

The discretization involves two sets of pressure and velocity variables. The first set corresponds to the explicit time level $t = t^k$, the second set corresponds to the implicit time level $t = t^{k+1}$. The same discretization pattern may be used twice: the first time with explicit variables, the second time with implicit variables. It is highly desirable to write a single function that acts differently depending on the time level

```
void do_crank_nickolson(Eq * eq)
{
    add_diffusive_term<Explicit>(eq);
    add_diffusive_term<Implicit>(eq);
}
```

Here, the `add_diffusive_term()` function is a *function template.* A time level appears as a template parameter that toggles explicit or implicit discretization. If the `TimeLevel` parameter is `Explicit`, all contributions should be made via the `add_explicit()` method; if the time level parameter is `Implicit`, all contributions should be made via `add_implicit()` method. This selective behavior is implemented via the `Contributor` class

```
template<typename TimeLevel> class Contributor;
```

`Contributor` has two *template specializations* corresponding to the implicit and explicit time levels

```
class Implicit{};
template<> class Contributor<Implicit>
{
public:
    static void add(Eq * eq, double k, size_t var, size_t i, size_t j)
    {
        eq->add_implicit(k, var, i, j);
    }
};


class Explicit{};
template<> class Contributor<Explicit>
{
public:
    static void add(Eq * eq, double k, size_t var, size_t i, size_t j)
    {
        eq->add_explicit(k, var, i, j);
    }
};
```

The `add_diffusive_term()` function uses `Contributor` to unify the implicit and explicit contributions made by the same discretization pattern

```
template<typename TimeLevel>
void add_diffusive_term(Eq * eq)
{
```

```
    typedef Contributor<TimeLevel> C;

    C::add( 1.0, U, i, j);
    C::add(-1.0, U, i+1, j);
}
```

Both `Contributor` specializations have a single trivial function, which is easily inlined by any up to date `C++` compiler. Since these functions are `static`, there is no need to create an object to call them. Consequently, the result of compilation of the above code is essentially the same as the result of compilation of the following code

```
void add_diffusive_term_explicit(Eq * eq)
{
    eq->add_explicit( 1.0, U, i, j);
    eq->add_explicit(-1.0, U, i+1, j);
}
void add_diffusive_term_implicit(Eq * eq)
{
    eq->add_implicit( 1.0, U, i, j);
    eq->add_implicit(-1.0, U, i+1, j);
}
```

Thus, there are no differences at run time. Of course, the latter code requires less compilation time. However, it demands maintenance of *two* functions while the former code has a single discretization routine. In our code this routine occupies more than 200 lines of code, and this is much easier to maintain compared to 400 lines of code split into two functions with slightly different implementations. The `Contributor` specializations are reused in the discretization of all terms in the governing equations and boundary conditions, which results in less maintenance with no runtime penalty.

There is another advantage of using `Contributor`. If only one of the two `add_diffusive_term` specializations is actually used, the second one *is not generated at all.* `C++` compilers are sophisticated enough to make a decision whether to generate a particular template specialization based on whether or not it is being used.

Thus, the generation and compilation of unused specializations are automatically suppressed.

3.4 Discretization of the vector equations

Equations corresponding to the velocity knots have two scalar components. In the previous Chapter, the unified (L, T) notation is introduced to provide a single explanation of discretization procedures which is valid for both components of the equations at the same time.

This approach can be expressed directly in the code with the use of C++. For each discretization pattern, a single discretization routine is created. Then, two mappings corresponding to the X and Y components are defined. Each discretization routine is a C++ template function that takes a mapping as a template parameter. The C++ compiler generates two discretization routines for both components automatically from this single template. This greatly reduces the size of the discretization code without any loss of performance as shown below.

The mappings are defined as C++ *traits* templates (see Alexandrescu (2001) and Vandevoorde and Josuttis (2002) for comprehensive studies of traits). The X and Y component traits are as follows

```
class XAxis
{
public:
    enum{l_axis = 1, t_axis = 2};

    typedef LeftWall Edge_l0;
    typedef RightWall Edge_l1;
```

```
    typedef BottomWall Edge_t0;
    typedef TopWall Edge_t1;

    static size_t l(size_t i, size_t j) {return i;}
    static size_t t(size_t i, size_t j) {return j;}
    static size_t i(size_t l, size_t t) {return l;}
    static size_t j(size_t l, size_t t) {return t;}
};

class YAxis
{
public:
    enum{l_axis = 2, t_axis = 1};

    typedef BottomWall Edge_l0;
    typedef TopWall Edge_l1;
    typedef LeftWall Edge_t0;
    typedef RightWall Edge_t1;

    static size_t l(size_t i, size_t j) {return j;}
    static size_t t(size_t i, size_t j) {return i;}
    static size_t i(size_t l, size_t t) {return t;}
    static size_t j(size_t l, size_t t) {return l;}
};
```

Each mapping defines the correspondence between the (i, j) and (l, t) indices as well as the correspondence between the left, right, bottom and top walls and the L_0, L_1, T_0, T_1 edges of the control volumes. All of the discretization routines accept axis traits as a template parameter. For the convective term, we have

```
template<typename TimeLevel, typename Axis>
void MomentumConvectiveTerm::get_discretization
    (DiscretizationEquation * eq, size_t i, size_t j)
{
    size_t l = Axis::l(i, j);
    size_t t = Axis::t(i, j);

    //edge L0
    add_convection_through_l_edge<TimeLevel, Axis>(eq, l, t, l-1);

    //edge L1
    add_convection_through_l_edge<TimeLevel, Axis>(eq, l, t, l+1);

    //edge T0
    add_convection_through_t_edge<TimeLevel, Axis>(eq, l, t, t-1);

    //edge T1
    add_convection_through_t_edge<TimeLevel, Axis>(eq, l, t, t+1);
}
```

Here, the convective term is discretized by taking fluxes through all four edges of the control volume into account, i.e., $U(i,j)$ if `Axis` is `XAxis`; and $V(i,j)$ if `Axis` is `YAxis`. Functions `add_convection_through_l_edge()` and `add_convection_through_t_edge()` are implemented using information provied by classes `XAxis` and `YAxis`. This information is processed during compile time to generate functions corresponding to the X and Y components from a single function template. Methods of classes `XAxis` and `YAxis` are inlined by any modern `C++` compiler. Since all template specializations are created at compile time, the code generated has the same optimization capacity as a nontemplate `C++` code. Note that the `get_discretization()` routine has two independent template parameters `TimeLevel` and `Axis` which allows it to be used in four different ways

```
void discretize(DiscretizationEquation * eq, size_t i, size_t j)
{
    //for control volume U(i, j)
    conv_term->get_discretization<Implicit, XAxis>(eq, i, j);
    conv_term->get_discretization<Explicit, XAxis>(eq, i, j);

    //for control volume V(i, j)
    conv_term->get_discretization<Implicit, YAxis>(eq, i, j);
    conv_term->get_discretization<Explicit, YAxis>(eq, i, j);
}
```

Thus, there is a single discretization routine instead of four routines with slightly different implementations, and there is no run time penalty in comparison to the code written by a programmer. The same technique is used for implementing all of the discretization routines corresponding to the velocity knots listed in Table 2.2.

3.5 Program architecture

Classes such as `PolyMatrixArray` and `DiscretizationEquation` represent abstract data types. The code which is written in terms of `PolyMatrixArray` and `DiscretizationEquation` uses these data types as elementary building blocks. Thus, such code is written at the higher level of abstraction than the code which implements these data types. A brief understanding of the structure of the code can be obtained by analyzing classes written at the highest level of abstraction. In this section, the key high-level classes are discussed. It is shown what these classes are responsible for, and how they interact with each other to achieve the desired functionality.

3.5.1 High-level classes

Solver

This is a central class which is equivalent to a conductor if the rest of the program is viewed as an orchestra. It creates all of the objects required during computations and provides access to information required by other objects to perform their computations. For example, it creates `PolyMatrixArray` which stores the values of the pressure and velocity variables, as well as two matrices that store fluid fractions in each computational cell at the explicit and implicit times. The memory required by these large objects is allocated *once* upon creation of the `Solver` object. The memory allocated is released in `Solver` destructor.

FluidAdvector

This class is responsible for the computation of a new time step estimation using the Courant-Friedrichs-Levy condition as well as advection of fluid fractions with local fluid velocity. It implements the advection part of the VOF method which is described in Section 2.2.

InterfaceReconstructor

This class corresponds to the reconstruction part of the VOF method. It provides information about the exact location of the fluid interface and computes fractional areas and volumes open to flow. Both the fluid-air and the fluid-solid interfaces are reconstructed with the help of two low-level classes `FreeSurfaceInterfaceReconstructor` and `BodyInterfaceReconstructor`, respectively. `InterfaceReconstructor` unifies these low-level classes and serves as a unique source of information about the fluid region geometry for the rest of the program. There are two objects of this class within a single `Solver` object. One of them corresponds to the implicit time and the other one corresponds to the explicit time.

CellClassificator

This class implements the classification of pressure cells and velocity knots which is described in Section 2.3. It provides information which is required to choose the discretization pattern for each computational knot.

SystemBuilder

This class applies the discretization routines to each pressure cell and velocity knot based on the knot type which is known from `CellClassificator`. Each discretization routine populates a `DiscretizationEquation` object, which is then used to fill up the next row in a sparse matrix and the free term of the linear system to be solved.

LinearSystem

`LinearSystem` stores the matrix and the free term column of the linear system. In addition, this class is responsible for the solution of the linear system. Currently, it uses the `AztecOO` library to solve the system using the generalized minimal residual method with ILUT preconditioner. The `AztecOO` library is a part of `Trilinos` project which is available under the conditions of the GNU Lesser General Public License v.2.1 at http://software.sandia.gov/trilinos/. The solution is placed directly into the `PolyMatrixArray` object that stores the pressure and velocity components.

3.5.2 Solver life cycle

The simulation of the unsteady flow is performed by an iterative call to the `Solver::life_cycle()` method, which is implemented by using classes described earlier

```
void Solver::life_cycle()
{
    dt = fluid_advector->advect_fluid(&vof_values);
    interface_rctor->reconstruct_interface(vof_values);
    cell_classificator->do_cell_classification(interface_rctor);
    system_builder->build_system(&system);
```

```
    system->solve();
    curr_time += dt;
}
```

Upon completion of the call to this method, `solver` has new values of the pressure and velocity stored in its `PolyMatrixArray` object. These values can be written to a file to store the simulation results.

It can be noted that the code functionality is divided among well-separated objects. Each object is responsible for a well-defined task. Each object accomplishes its task with the help of a number of low-level objects; the same is true for non-trivial low-level objects as well. This is another application of the well-known "divide et impera" principle, which results in significant simplifications in writing, understanding and maintaining object-oriented programs (see, for example, Booch (1994)).

Due to the creative use of expressiveness of the `C++` programming language, the code of the `life_cycle()` function corresponds directly to the simulation workflow description provided in Section 2.9.

3.6 Validation of uniform flow simulations

The numerical simulation tool developed in this thesis can be applied to a class of free surface problems with arbitrarily moving rigid bodies. Here this numerical simulation tool is applied specifically to the problem of unsteady, laminar, two-dimensional flow of a viscous incompressible fluid past a transversely oscillating cylinder in the presence of a free surface. These results and their analysis will be presented in the

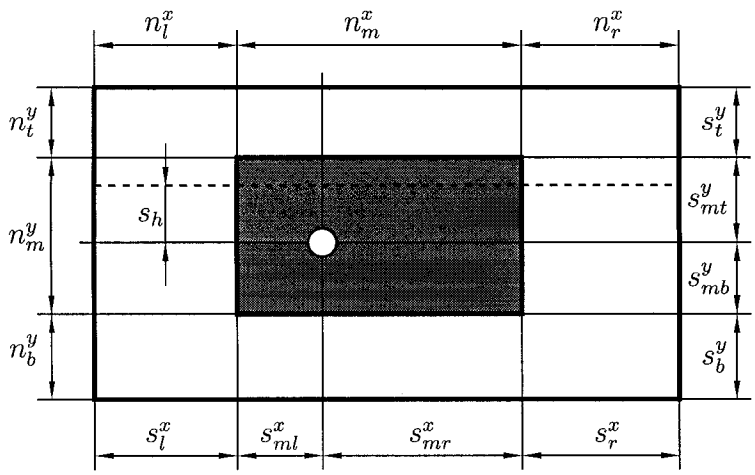

Figure 3.1: The computational grid. The uniform grid region is shaded.

next Chapter. In this Chapter, the validation of the numerical simulation tool is presented for uniform flow past (i) a stationary cylinder (no forced oscillations); (ii) a cylinder undergoing forced rotational oscillations and (iii) a cylinder undergoing forced transverse oscillations in the absence of a free surface, in addition to uniform flow past a stationary cylinder in the presence of a free surface.

3.6.1 Grid specification

Before presenting the numerical simulation results, there are several parameters of the grid geometry which must be specified. The computational grid used for numerical investigation of flow around a circular cylinder is shown in Figure 3.1. The computational domain geometry is defined with respect to the *mean* position of the cylinder. In the non-stationary case, the cylinder is moving or oscillating around its mean position. In the vicinity of the mean cylinder position, the grid has fine reso-

lution and is uniform. The uniform grid region is shaded in Figure 3.1. Outside of the shaded region, the grid expands exponentially towards the four boundaries of the computational domain. The computational domain dimensions are defined by specifying scale parameters s_l^x, s_{ml}^x, s_{mr}^x, s_r^x along the X axis and s_b^y, s_{mb}^y, s_{mt}^y, s_t^y along the Y axis. Physical dimensions are obtained by multiplying these parameters by the cylinder diameter. The initial elevation of the fluid is specified by s_h. To consider the case of uniform flow past a cylinder in the absence of a free surface, it is sufficient to define s_h so that it is greater than $s_{mt}^y + s_t^y$.

The values of n_l^x, n_m^x and n_r^x specify the number of knots in the left, middle, and right areas of the computational grid, respectively. The values of n_b^y, n_m^y and n_t^y specify the number of knots at the bottom, middle, and top areas of the computational grid, respectively. Grid steps in the uniform grid region h_m^x and h_m^y are computed as

$$h_m^x = \frac{s_{ml}^x + s_{mr}^x}{n_m^x - 1} d, \qquad h_m^y = \frac{s_{mb}^y + s_{mt}^y}{n_m^y - 1} d. \tag{3.2}$$

The exponential expansion of the grid is defined as follows. Let $\{\xi_i\}_{i=1}^n$ be a uniform distribution of n knots in the interval $[0, 1]$

$$\xi_i = \frac{i-1}{n-1}, \quad i = 1 \ldots n. \tag{3.3}$$

The exponential distribution of n knots $\{x_i\}_{i=1}^n$ in the interval $[x_a, x_b]$ is then given by

$$x_i = x_a + (x_b - x_a)\frac{e^{a\xi_i} - 1}{e^a - 1}, \qquad i = 1 \ldots n. \tag{3.4}$$

The parameter a is chosen so that the first grid step of the exponential grid at the

end attached to the uniform grid region is equal to the uniform grid step h_m^x. This is done by finding the root of the following function

$$\mathcal{F}(a) = h_m^x - (x_b - x_a)\frac{e^{a/(n-1)} - 1}{e^a - 1}. \tag{3.5}$$

The exponential grid parameter, a_0, satisfying $\mathcal{F}(a_0) = 0$ is determined numerically using Brent's algorithm (see Brent (1973)). Four exponential grid parameters corresponding to the left, right, bottom, and top parts of the computational domain are computed automatically. Thus, the parameters shown in Figure 3.1 are sufficient to describe the grid specification completely.

3.6.2 Choice of computational parameters

Numerical solution depends on a number of computational parameters such as the time step, Δt, and the grid geometry parameters: s_l^x, s_{ml}^x, s_{mr}^x, s_r^x, s_b^y, s_{mb}^y, s_{mt}^y, s_t^y, n_l^x, n_m^x, n_r^x, n_b^y, n_m^y, n_t^y, and s_h. A large number of numerical experiments are performed to choose the values of the grid parameters. It is observed that numerical solutions obtained using computational domain sizes larger than $s_l^x = 10$, $s_r^x = 20$, $s_b^y = s_t^y = 15$ differ only slightly from each other. For example, in the case of uniform flow past a stationary cylinder in the absence of a free surface the results obtained are accurate for Strouhal numbers to about 2.5%. These grid parameters are found to be sufficient to allow us to neglect the computational domain boundary effects. The grid parameters used in this thesis are consistent with the grid parameters used in the studies of flow past cylinders conducted by other researchers (see, for example,

Reichl (2001) and Reichl *et al.* (2005)).

To test the influence of the size of the time step, numerical simulations corresponding to two values of the time step, 10^{-2} and 5×10^{-3}, were performed. The difference in the peak values of the lift force coefficient obtained from numerical simulations with these time steps vary within 1.6%; the difference between Strouhal numbers is found to be 1%. This difference is considered to be acceptable for the present thesis. The time step of 10^{-2} is chosen since it reduces the actual time required to perform the numerical simulations.

The grid resolution near the cylinder boundary and in the near wake region is critical. This is determined by the values of grid scale parameters s^x_{mr}, s^x_{ml}, s^y_{mb}, s^y_{mt} as well as the knot numbers n^x_m and n^y_m. The grid resolution can be expressed in terms of the number of grid cells per cylinder diameter. The structured grid used in this thesis is less flexible compared to unstructured grids. Computational knots cannot be added individually at the desired points in space using such a grid. Only an entire row or column of knots may be added in order to increase the grid resolution near the cylinder. Adaptive grid refinement suggested by Berger and Oliger (1984) can be used to overcome this drawback of structured grids, but this is not implemented in the present work since it would add extra complexity to the problem. From the numerical experiments conducted with different grid resolutions, it is found that setting the number of cells per cylinder diameter to 20 - 30 is sufficient to obtain reasonably accurate results.

3.6.3 Flow past a stationary circular cylinder

To verify the accuracy of the present algorithm, numerical results are first obtained at small times and compared to the results from the numerical methods of Li *et al.* (2004) and Ploumhans and Winckelmans (2000). Li *et al.* implement the lattice-Boltzmann method to solve the problem of impulsively started flow past a circular cylinder for a Reynolds number of R = 550. Ploumhans and Winckelmans use a two-dimensional vortex method based on the vorticity-stream function formulation of the Navier-Stokes equations in combination with the particle strength exchange scheme for diffusion. The values of the computational parameters used in the present numerical simulation are specified in Table 3.1.

Δt	s_l^x	s_{ml}^x	s_{mr}^x	s_r^x	n_l^x	n_m^x	n_r^x	s_b^y	s_{mb}^y	s_{mt}^y	s_t^y	n_b^y	n_m^y	n_t^y
10^{-3}	20	1	1	40	70	80	140	17	1	1	17	60	80	60

Table 3.1: The values of the computational parameters corresponding to the present numerical results shown in Figures 3.2 and 3.3.

The development of the vorticity at small values of time is compared with the results of Li *et al.* (2004) in Figure 3.2. This figure shows that the present numerical solution accurately calculates the size, location, and the shape of the developing vortices. In addition, corresponding to an impulsive start, the initial formation of vortices behind the cylinder is observed, with the increase of time, a second vortex (at $t = 1.5$) develops and then interacts with the original vortex. The resulting drag coefficient calculation is displayed in Figure 3.3. The result which is compared with that obtained by Li *et al.* (2004) and Ploumhans and Winckelmans (2000) shows

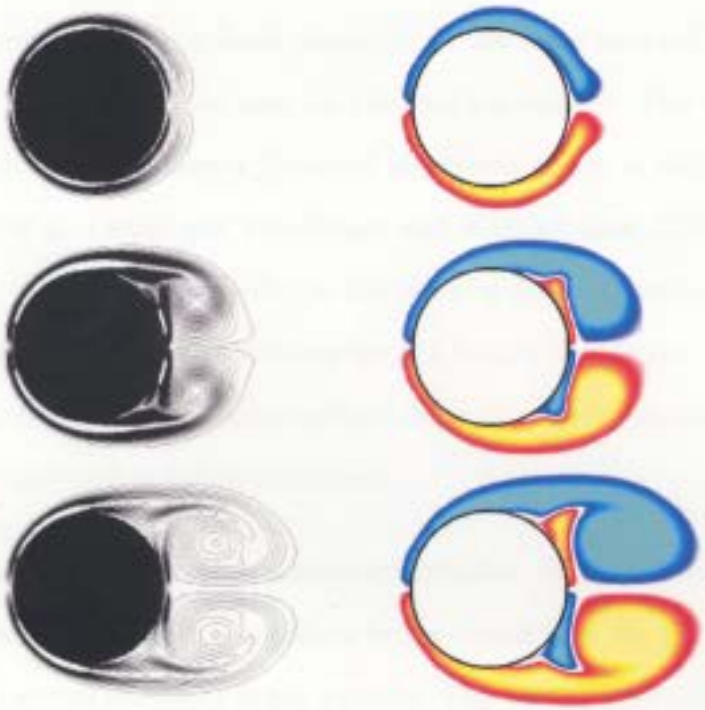

Figure 3.2: Equivorticity patterns for the case of uniform flow past a stationary cylinder in the absence of a free surface at R = 550: Li *et al.* (2004) (left), present work (right). $t = 0.5$ (top), $t = 1.5$ (middle), $t = 2.5$ (bottom).

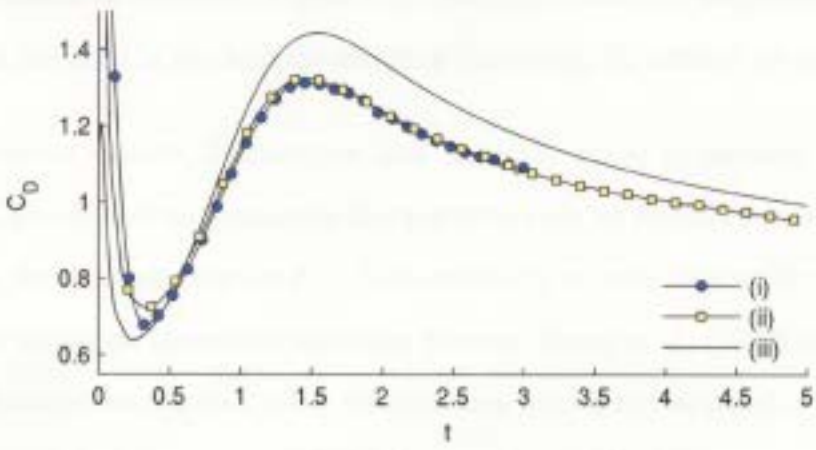

Figure 3.3: Early time development of the drag force coefficient, C_D, for uniform flow past a stationary cylinder in the absence of a free surface at R = 550: (i) - Li *et al.* (2004), (ii) - Ploumhans and Winckelmans (2000), (iii) - present work.

a good agreement. There is a local minimum in the time interval between 0.2 and 0.4 and a local maximum in the interval between 1.4 and 1.6. The value of the drag coefficient at the local maximum observed in present study is higher than the one obtained by Li *et al.* (2004) and Ploumhans and Winckelmans (2000). It is believed that the higher drag coefficient is due to the use of different boundary conditions: Li *et al.* use periodic boundary conditions for the lattice-Boltzmann equations at the upper and lower boundaries of computational domain, whereas inviscid wall boundary conditions are employed in the present study.

In the case of uniform flow past a stationary cylinder, the flow only depends on the Reynolds number. For Reynolds numbers below about $R \approx 40$, the cylinder wake is stable and thus vortex shedding is not present. The vortices, formed in the separate layer, remain attached to the cylinder surface. As the Reynolds number increases, however, there is a transition from the formation of the attached eddies to the shedding of vortices from alternate sides of the cylinder, creating the classical Kármán vortex street as shown in Figure 3.4. The dimensionless frequency at which these vortices are shed is the natural shedding frequency, f_0, defined by equation (1.23).

In numerical studies, fluctuations that naturally occur in physical experiments are absent, which leads to symmetric flow patterns even for Reynolds numbers higher than 40 (see, for example, Figure 3.2). It is necessary to introduce artificial perturbations in order to trigger the vortex shedding process. Braza *et al.* (1986) showed that these perturbations are required solely for switching from a symmetrical to an asymmetrical flow regime, and the resulting flow is independent of the particular perturbation that caused it. Braza *et al.* suggest to apply a perturbation by rotating the cylinder for

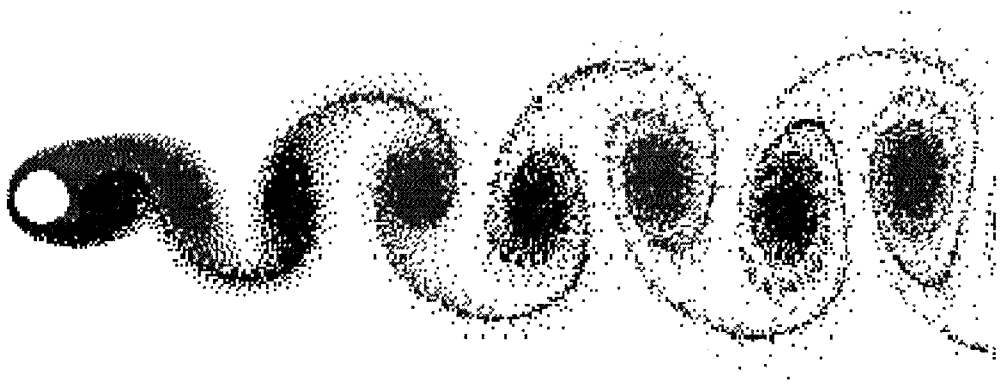

Figure 3.4: Kármán vortex street downstream of a circular cylinder (Cesareo de La Rosa Siqueira (2006)).

a short period of time at the beginning of simulation. This is the approach used in this thesis.

The periodic nature of the flow is reflected in the lift and drag force coefficients. The dimensionless vortex shedding frequency, f_0, (which is also known as the natural Strouhal number St) can be computed by calculating the power spectral density (PSD) of the lift coefficient C_L through Fourier analysis. Table 3.2 shows the values of the mean drag coefficient, $\overline{C_D}$, the mean peak values of the absolute value of the lift coefficient, $\widehat{C_L}$, and the predicted value of the natural vortex shedding frequency, f_0, for four values of the Reynolds number in the range $200 \leqslant \mathrm{R} \leqslant 500$. The values of the computational parameters used in the numerical simulations performed to obtain results shown in Table 3.2 are specified in Table 3.3. From Table 3.2 it can be observed that $\overline{C_D}$, $\widehat{C_L}$ and f_0 agree very well for $\mathrm{R} = 200$. $\widehat{C_L}$ remains well-predicted even for higher Reynolds numbers, while the $\overline{C_D}$ values obtained in the present study

R	$\overline{C_D}$			$\widehat{C_L}$		f_0		
	(i)	(ii)	(iii)	(i)	(iii)	(i)	(ii)	(iii)
200	1.3389	1.3412	1.34	0.70	0.72	0.199	0.1972	0.194
300	1.3820	1.3769	1.41	0.96	0.97	0.211	0.2113	0.207
400	1.4080	1.4142	1.50	1.08	1.13	0.2228	0.2198	0.214
500	1.4433	1.4449	1.54	1.23	1.23	0.230	0.2254	0.218

Table 3.2: The predicted mean drag coefficient, $\overline{C_D}$, mean peak values of the absolute value of the lift coefficient, $\widehat{C_L}$, and the natural shedding frequency, f_0, compared with previous numerical studies for the case of a stationary cylinder. (i) - Poncet (2004), (ii) - Henderson (1997), (iii) - present work.

Δt	s_l^x	s_{ml}^x	s_{mr}^x	s_r^x	n_l^x	n_m^x	n_r^x	s_b^y	s_{mb}^y	s_{mt}^y	s_t^y	n_b^y	n_m^y	n_t^y
10^{-2}	15	1	3	30	40	160	80	15	1	1	15	40	80	40

Table 3.3: The values of the computational parameters corresponding to the present numerical results shown in Table 3.2.

are higher than the computed $\overline{C_D}$ values by Poncet (2004) and Henderson (1997). The predicted values of the vortex shedding frequency are somewhat smaller due to numerical viscosity introduced by the relatively low-order scheme employed in the present study. This artificial diffusion results in a flow behaving as if it were at a slightly lower Reynolds number, which is consistent with smaller values observed in Table 3.2.

3.6.4 Flow past a circular cylinder under forced rotational oscillation

For the case of uniform flow past a circular cylinder undergoing forced rotational oscillation numerical simulations are carried out at R = 100. The values of the computational parameters used in the present numerical simulation are specified in Table 3.4. The angular cylinder displacement is taken to be

$$a(t) = -\frac{\Omega}{\pi f}\cos(2\pi f t). \tag{3.6}$$

Here, f is the dimensionless frequency of rotational cylinder oscillation and Ω is the dimensionless maximum rotational speed given by

$$\Omega = \frac{\dot{a}_{\max} d}{2U}, \tag{3.7}$$

where $\dot{a}_{\max}$ is the maximum angular velocity. The flow is then uniquely defined by specifying the values of the dimensionless parameters Ω, f and the Reynolds number, R. In Figure 3.5, the present predictions of the equivorticity patterns for the rotationally oscillating cylinder immersed in a uniform flow are compared with those obtained from the numerical study of Choi *et al.* (2002) at R = 100 for four different flow regimes. Choi *et al.* employed a finite volume method to solve the Navier-Stokes equations in their pressure-velocity formulation on a polar O-type grid. It can be seen that these results are in good agreement.

Δt	s_l^x	s_{ml}^x	s_{mr}^x	s_r^x	n_l^x	n_m^x	n_r^x	s_b^y	s_{mb}^y	s_{mt}^y	s_t^y	n_b^y	n_m^y	n_t^y
10^{-2}	12	3	5	20	30	180	50	12	4	4	12	30	180	30

Table 3.4: The values of the computational parameters corresponding to the present numerical results shown in Figure 3.5.

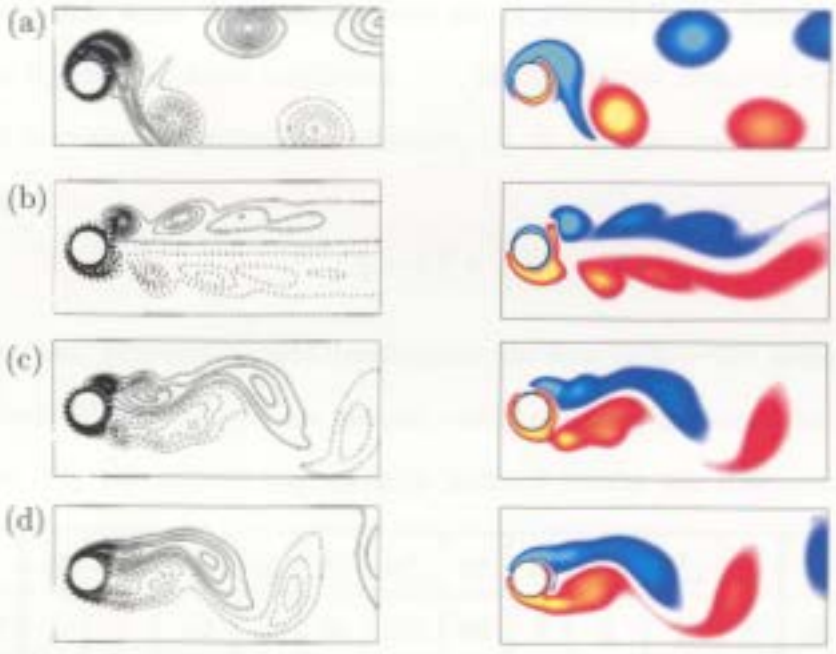

Figure 3.5: Equivorticity patterns corresponding to four different flow regimes at R = 100: (a) $\Omega = 2$, $f = 0.165$; (b) $\Omega = 2$, $f = 0.4$; (c) $\Omega = 2$, $f = 0.8$; (d) $\Omega = 0.6$, $f = 0.8$. Choi *et al.* (left), present work (right).

3.6.5 Flow past a circular cylinder under forced transverse oscillation

For the case of uniform flow past a circular cylinder undergoing forced transverse oscillation numerical simulations are carried out at R = 200. The values of the computational parameters used in the present numerical simulation are specified in

Table 3.5. The transverse cylinder displacement is taken to be

$$y(t) = -A\cos(2\pi ft). \tag{3.8}$$

It is noted that when the computations are performed in the non-inertial frame of reference $\mathbb{X}_g$, the lift force coefficient, C_L, in the inertial frame of reference can be obtained from the computed lift coefficient, C_L^g, in the non-inertial frame of reference according to

$$C_L = C_L^g + \frac{\pi}{2}\ddot{y}(t). \tag{3.9}$$

Meneghini and Bearman (1995) investigated the flow at R = 200 using the vortex in cell method. They predicted the natural vortex shedding frequency to be $f_0 = 0.196$. In Figure 3.6, the present equivorticity pattern results are displayed for the case

Δt	s_l^x	s_{ml}^x	s_{mr}^x	s_r^x	n_l^x	n_m^x	n_r^x	s_b^y	s_{mb}^y	s_{mt}^y	s_t^y	n_b^y	n_m^y	n_t^y
10^{-2}	5	1	10	10	30	150	60	6	2	2	6	36	60	36

Table 3.5: The values of the computational parameters corresponding to the present numerical results shown in Figure 3.6.

$R = 200$: $f/f_0 = 0.8$ and $A = 0.6$ and compared with those obtained by Meneghini and Bearman (1995). The comparison between these results shows a good agreement in the near wake region.

The flow is also calculated at R = 855, $A = 0.13$ in the frequency ratio range $0.5 \leqslant f/f_0 \leqslant 4.0$ and compared with the experimental results of Ongoren and Rockwell (1988) and numerical results of Al-Mdallal (2004). The values of the computational parameters used in the present numerical simulation are specified in Table 3.6.

Figure 3.6: Equivorticity patterns in the case of a transversely oscillating cylinder at R = 200, $f/f_0 = 0.8$, $A = 0.6$. Meneghini and Bearman (1995) (left), present work (right).

Good qualitative agreement is obtained between these results as shown in Figure 3.7.

Δt	s_l^x	s_{ml}^x	s_{mr}^x	s_r^x	n_l^x	n_m^x	n_r^x	s_b^y	s_{mb}^y	s_{mt}^y	s_t^y	n_b^y	n_m^y	n_t^y
10^{-2}	12	3	7	20	30	180	50	12	4	4	12	30	144	30

Table 3.6: The values of the computational parameters corresponding to the present numerical results shown in Figure 3.7.

3.6.6 Free surface flow past a stationary circular cylinder

For the case of uniform flow past a stationary cylinder in the presence of a free surface numerical simulations are carried out at R = 180 and in the range of Froude numbers $0 \leqslant \mathrm{Fr} \leqslant 0.4$ when the depth of cylinder submergence h is 0.55. The values of the computational parameters used in the present numerical simulation are specified in Table 3.7. Figure 3.8 shows a comparison between the equivorticity patterns obtained in the present study and those obtained by Reichl *et al.* (2005). Reichl *et al.* use the commercial CFD software package FLUENT which is based on the finite volume method

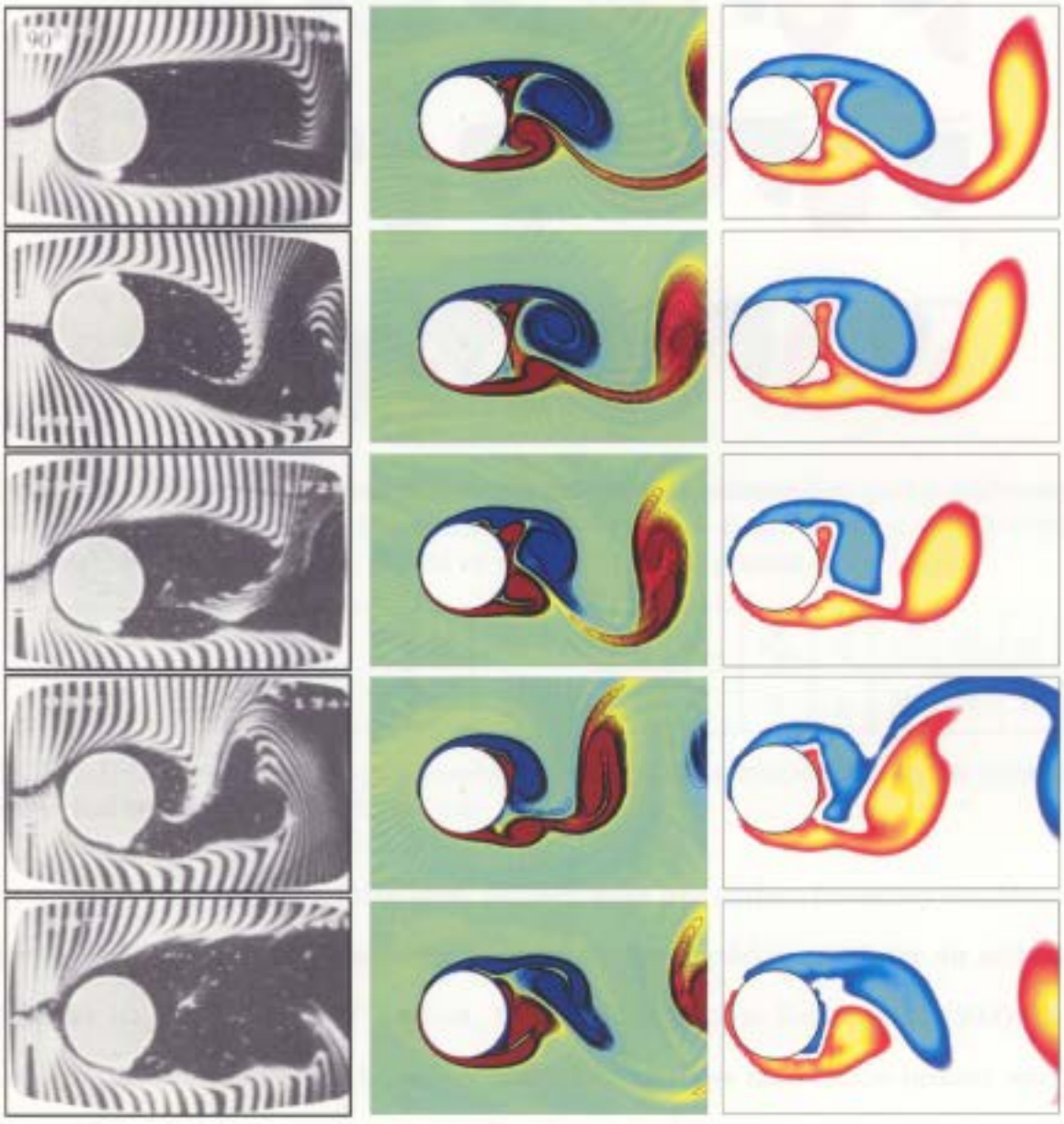

Figure 3.7: Comparison of flow visualization by Ongoren and Rockwell (1988) (left), equivorticity patterns by Al-Mdallal (2004) (middle) and present work (right) for uniform flow past a transversely oscillating cylinder at R = 855, $A = 0.13$ and $f/f_0 = 0.5, 1.0, 2.0, 3.0, 4.0$ (from top to bottom).

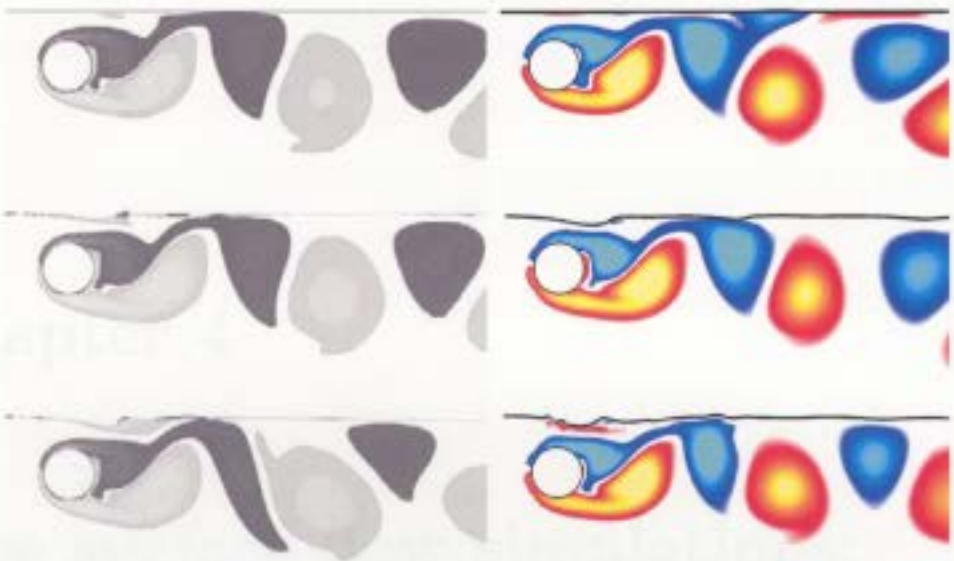

Figure 3.8: Comparison of the equivorticity patterns for uniform flow past a stationary cylinder in the presence of a free surface at R = 180, $h = 0.55$: Fr = 0 (top), Fr = 0.3 (middle), Fr = 0.4 (bottom). Reichl *et al.* (2005) (left), present work (right).

Δt	s_l^x	s_{ml}^x	s_{mr}^x	s_r^x	n_l^x	n_m^x	n_r^x	s_b^y	s_{mb}^y	s_{mt}^y	s_t^y	n_b^y	n_m^y	n_t^y
5×10^{-3}	5	1	7	3	30	120	18	5	3	2	1	50	75	15

Table 3.7: The values of the computational parameters corresponding to the present numerical results shown in Figure 3.8.

on a structured, boundary fitted grid. In their work free surface boundary conditions are applied implicitly by considering a two-phase flow model in which the air and the fluid are tracked by the VOF method. Bearing in mind that Reichl *et al.* (2005) did not use the flow model and boundary conditions as those used in the present work, it may be noted that, even so, there is a good qualitative agreement between the results.

Chapter 4

Free surface flow simulations: Forced transverse oscillations of a circular cylinder

This Chapter focuses on the results for flow past a transversely oscillating cylinder in the presence of a free surface. The numerical simulations are conducted at a Reynolds number of R = 200 for a fixed submergence depth, $h = 1.25$, and displacement amplitudes $A = 0.25$ and 0.5 in the frequency ratio range $0.95 \leqslant f/f_0 \leqslant 4.0$. The results are analyzed by means of equivorticity patterns in the near wake region as well as the fluid forces acting on the cylinder. The character of the vortex shedding is important as it influences the phase of lift force and, consequently, the mechanical energy transfer between the fluid and the body. It is noted that the classical definition of locked-on flow regimes is based on the existence of a dominant peak in the PSD of

the lift coefficient. However, the determination of locked-on flow is not confined to the lift records. For example, Anagnostopoulos (2000) uses the PSD analysis or repetition of the velocity traces to classify locked-on modes and Ongoren and Rockwell (1988) classify lock-on regimes based on a repetition of vortex shedding in the near wake region over an integer number of cylinder oscillation periods. Williamson and Roshko (1988) explored the existence of locked-on vortex modes in the wavelength-amplitude plane for the case of a transversely oscillating cylinder in the absence of a free surface (symbolically represented by $h = \infty$). In their work, the Reynolds number is kept within a certain range $300 < R < 1000$ and the wavelength ratio is defined as $\lambda/d = UT/d$, where $T = 1/f$ is the period of oscillation (i.e., R is never held fixed and if the parameters of the present problem characterizing the periodic motion are introduced then $0 < A \leqslant 1.8$ and $f/f_0 \geqslant 0.3$). The major vortex patterns near the fundamental lock-on region ($\lambda/d \approx 5$ or $f \approx f_0$) are the three basic asymmetric modes: **2S**, **2P** and **P+S**, where **S** and **P** indicate 'single' and 'pair' of shed vortices, respectively. For the **2S** mode a single vortex of opposite rotation is alternatively shed from each side of the cylinder over a single vortex shedding cycle, T_v, defined by $T_v = kT$, where k is either a fractional or an integer number. The **2S** mode is known as the classical Kármán vortex street type of mode. The **2P** mode corresponds to the shedding of two counter-rotating pairs of vortices from each side of the cylinder per T_v. The **P+S** mode indicates a counter-rotating vortex pair is shed from one side of the cylinder and a single vortex is shed from the other side over one cycle, T_v. A map of these locked-on vortex patterns near the fundamental lock-on region observed by Williamson and Roshko (1988) is shown in Figure 4.1. In this thesis, the vortex formation modes are categorized using the terminology of Williamson and Roshko.

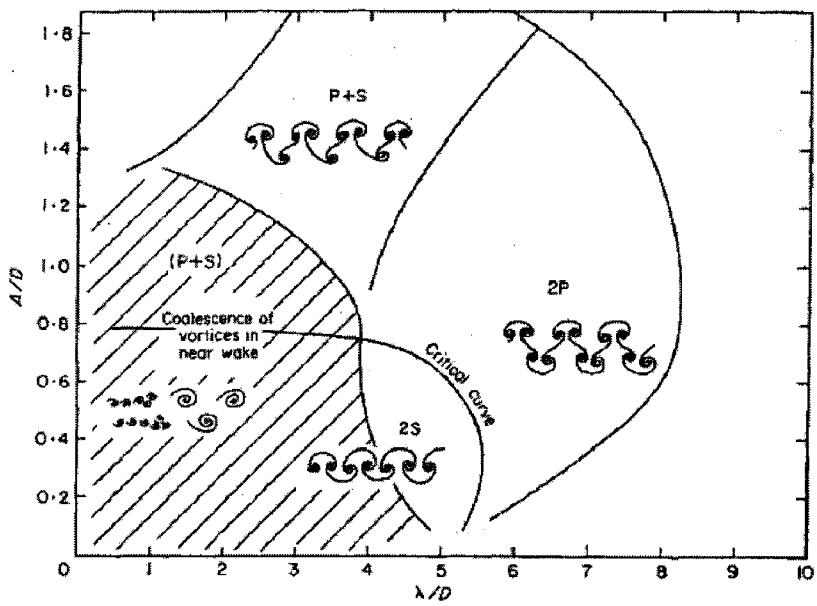

Figure 4.1: A map of the vortex lock-on patterns near the fundamental lock-on region observed by Williamson and Roshko. The critical curve marks the transition from one mode of vortex formation to another (Williamson and Roshko (1988), p. 362).

Following the methodology of Cetiner and Rockwell (2001), the Lissajous representations of lift and drag coefficients are used to demonstrate the mechanical energy transfer, degree of phase-locking, or a loss of lock-on and associated phase shift. The Lissajous curve describes the time change of lift or drag coefficient as a function of cylinder displacement. The relation between the alterations of flow patterns and the unsteady fluid forces on the cylinder surface is discussed. An analysis of the vortex shedding modes in the near wake region via Lissajous patterns and PSD analysis of unsteady lift coefficients is also included. In PSD analysis the time dependent traces of lift and drag coefficients are decomposed into the sum of sine functions using Fourier

analysis. This analysis allows the representation of a spectrum of the original function as a set of frequencies with which this function oscillates. Particular harmonics in the spectra suggesting lock-on in the near wake are represented by sharply defined peaks at integer and fractional values of the natural shedding frequency, f_0. The numerical parameters used in the present investigation are summarized in Table 4.1. It is noted that for the free surface flow calculations at $f/f_0 = 3.0$ and 4.0, the time step, Δt, is reduced to 2.5×10^{-3} and 5×10^{-3}, respectively. The predicted natural

Δt	s_l^x	s_{ml}^x	s_{mr}^x	s_r^x	n_l^x	n_m^x	n_r^x	s_b^y	s_{mb}^y	s_{mt}^y	s_t^y	n_b^y	n_m^y	n_t^y
10^{-2}	12	3	7	20	30	180	50	12	4	4	2	30	144	5

Table 4.1: The values of the computational parameters corresponding to the present investigation.

vortex shedding frequency at R = 200 is $f_0 = 0.191$ which is calculated using the PSD analysis of the lift coefficient for the case of uniform flow past a stationary cylinder in the absence of a free surface.

The present numerical simulations are carried out on `AMD Athlon 3000+` computers located at the Department of Mathematics and Statistics, Memorial University of Newfoundland. The unsteady flow calculations are conducted for large values of the time up to $t = 120$, using the computational parameters shown in Table 4.1. This requires about 48 hours of computational time.

The observed features are compared with the case of a transversely oscillating cylinder in the absence of a free surface. A series of forced transverse cylinder oscillations in a steady uniform flow in the absence of a free surface is analyzed under the same oscillation conditions to better understand what differences result from the inclusion

of the free surface located at a submergence depth of $h = 1.25$.

4.1 Vortex formation modes and fluid forces: R = 200, $A = 0.25$ and $h = 1.25$

The case of R = 200, $A = 0.25$ and $h = 1.25$ is first considered. At this low amplitude $A = 0.25$, the transverse cylinder oscillation produces a modified form of the classical Kármán vortex street type mode (**2S** mode) in all frequency ratios considered in $0.95 \leqslant f/f_0 \leqslant 4.0$ for $h = 1.25$ as well as in the case when a free surface is absent. Unless otherwise indicated, in all equivorticity plots that follow, the snapshots are taken at the instant $y(t) = A$ and every quarter oscillation cycle thereafter marked by the dots on the lift curves. The positive equivorticity patterns are indicated by blue coloured (clockwise rotating) contours and negative ones by red coloured (counter-clockwise rotating) contours. The vortex shedding in the near wake is locked-on over one period of cylinder oscillation for $f/f_0 = 0.95$ and 1.0. In Figure 4.2, typical equivorticity patterns as well as the lift coefficient display a periodic pattern over one period of cylinder oscillation, T, at $f/f_0 = 1.0$: $h = 1.25$ and ∞. The snapshots at $t = 0T$ (top) and $t = T$ (bottom) are nearly identical. Therefore, the locked-on asymmetric **2S** mode per T is observed for $f/f_0 = 1.0$. Comparison of Figure 4.2(a) with Figure 4.2(b) indicates that the free surface presence at $h = 1.25$ increases the vortex formation length about 12.9% at $f/f_0 = 1.0$. The shed vortex becomes weaker as it moves downstream due to its interaction with the free surface interface. The positive vorticity reaches its maximum amount at the instants $t = 0T$ and $t = T$;

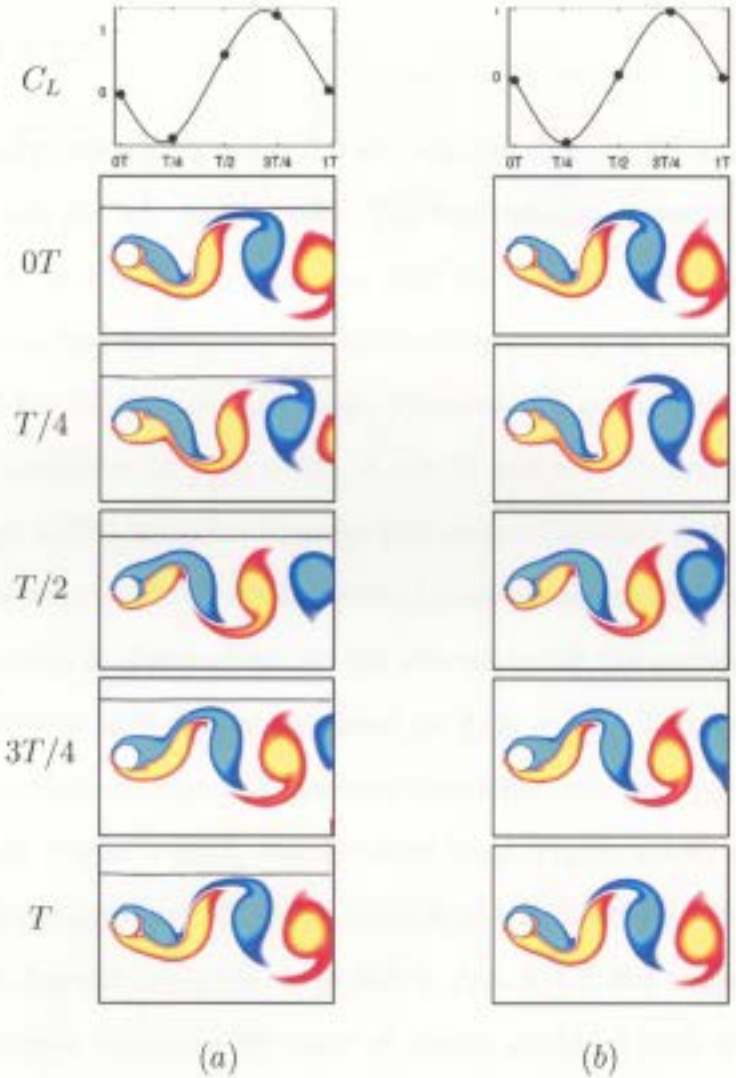

Figure 4.2: Equivorticity patterns over one period of cylinder oscillation, T, for $R = 200$, $A = 0.25$ and $f/f_0 = 1.0$ ($T \approx 5.2$, $106.6 \leqslant t \leqslant 111.8$): (a) $h = 1.25$ and (b) $h = \infty$. The instantaneous snapshots are marked in the corresponding lift coefficient curves shown above the snapshots.

whereas the negative vorticity reaches its maximum amount at the instant $t = T/2$. When the fully developed positive vortex propagates from the cylinder, a negative peak occurs on the C_L trace at $t \approx T/4$. Similarly, development and then propagation of the fully developed negative vortex results in the occurance of a positive peak on

the C_L trace at $t \approx 3T/4$.

At higher frequency ratios $f/f_0 = 2.0$, 3.0 and 4.0, the classical **2S** mode is obtained per $2T$, per $3T$ and per $4T$, respectively. The equivorticity patterns at $f/f_0 = 4.0$ are plotted over $4T$ in Figure 4.3. It is clear that the cylinder oscillation in this case produces a phase-locked pattern for the lift coefficients over $4T$. The snapshots at $t = 0T$ (top) and $t = 4T$ (bottom) are nearly identical. Thus, the lock-on occurs over four periods of oscillation at $f/f_0 = 4.0$: $h = 1.25$ and ∞. Comparison of Figure 4.3(a) with Figure 4.3(b) indicates that the free surface presence at $h = 1.25$ results in a significant increase ($\approx 50\%$) in the vortex formation length. In addition, a large free surface distortion is observed due to the interaction of the vortex with the free surface at a relatively high oscillatory speed at $f/f_0 = 4.0$. This seems to speed up the vortex shedding process and produces exact periodic flow patterns over $4T$ when $h = 1.25$ (see Figure 4.3(a)). On the other hand, Figure 4.3(b) displays quasi-periodic vorticity patterns over $4T$ in the absence of a free surface. Thus, as a result of increasing f/f_0 beyond 1.0 in the range $0.95 \leqslant f/f_0 \leqslant 4.0$, the near-wake vorticity produces the common locked-on **2S** mode of vortex shedding with different vortex shedding periods for $h = 1.25$. Period doubling, tripling and quadrupling relative to the classical Kármán mode (**2S** mode per T: $f/f_0 = 0.95$ and 1.0), occurs in the cases $f/f_0 = 2.0$, 3.0, 4.0, respectively. Also, about 50% increase in the vortex formation length is observed as f/f_0 increases from 0.95 to 4.0 for $h = 1.25$. However, a 40% decrease in the vortex formation length is observed as f/f_0 increases from 0.95 to 4.0 in the absence of a free surface. Thus, the effect of increasing f/f_0 in the absence of a free surface is to suppress the vortex shedding process.

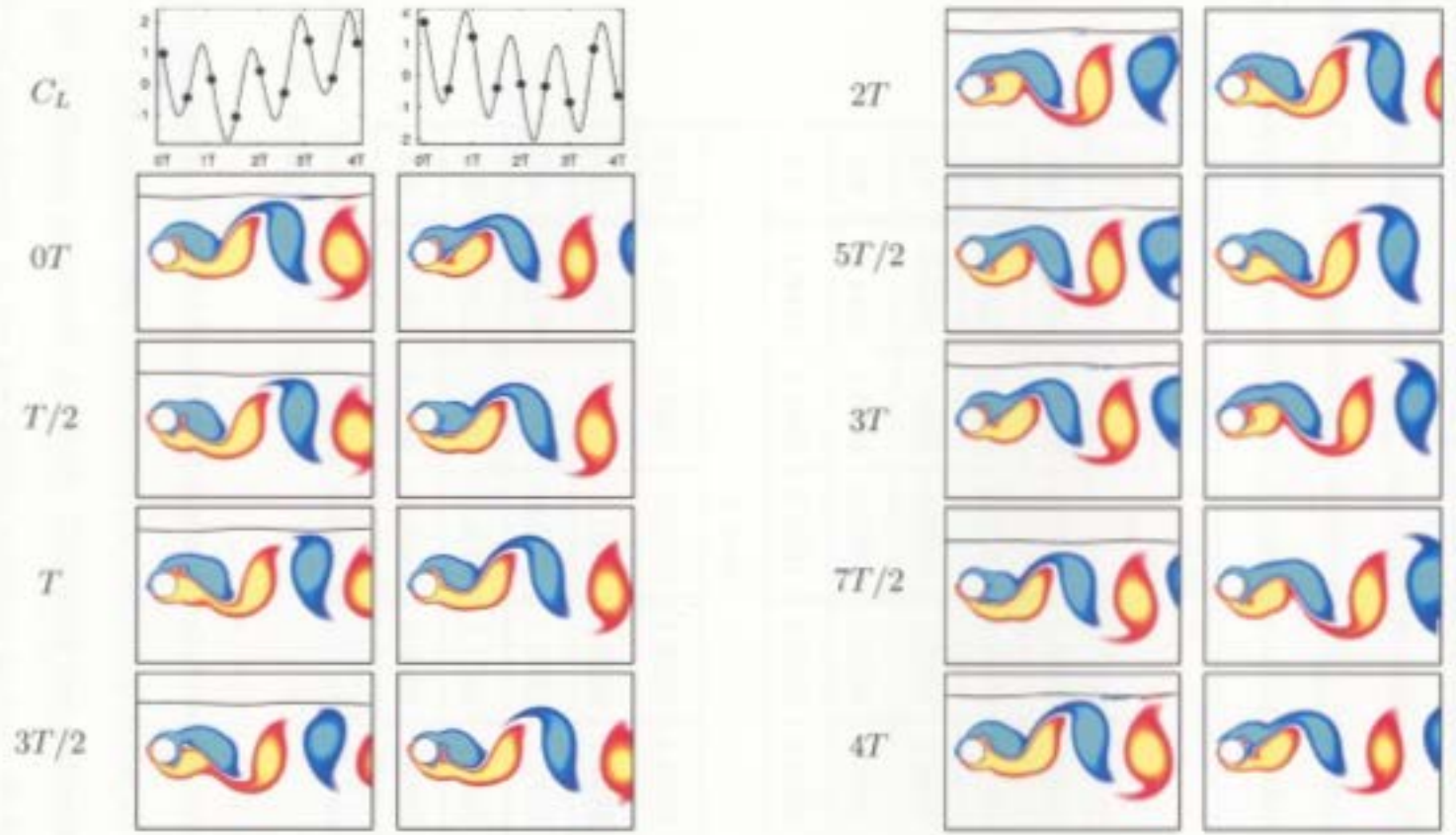

Figure 4.3: Equivorticity patterns over four periods of cylinder oscillation, $4T$, for R = 200, $A = 0.25$ and $f/f_0 = 4.0$ ($T \approx 1.3$, $96.8 \leqslant t \leqslant 102$): (a) $h = 1.25$ and (b) $h = \infty$. The instantaneous snapshots are marked in the corresponding lift coefficient curves shown above the snapshots.

Table 4.2 summarizes the effect of f/f_0 and h on the minimum lift coefficient, $C_{L_{min}}$, the maximum lift coefficient, $C_{L_{max}}$, the mean lift coefficient, $\overline{C_L}$; the minimum drag coefficient, $C_{D_{min}}$, the maximum drag coefficient, $C_{D_{max}}$ and the mean drag coefficient, $\overline{C_D}$. Comparison of these results in the presence of free surface located at $h = 1.25$

$h = 1.25$						
f/f_0	$C_{L_{min}}$	$C_{L_{max}}$	$\overline{C_L}$	$C_{D_{min}}$	$C_{D_{max}}$	$\overline{C_D}$
0.95	-0.7516	1.2929	0.2755	1.5803	1.9982	1.8020
1.0	-0.8437	1.3698	0.2742	1.6216	2.1901	1.9191
2.0	-0.6076	1.4683	0.3926	1.5215	2.4857	1.7954
3.0	-1.1404	1.8043	0.2598	1.3538	2.7209	1.8986
4.0	-1.8447	2.4468	0.2561	1.0779	2.8750	1.8082
$h = \infty$						
0.95	-0.9399	0.9404	-0.0032	1.5387	1.9739	1.7585
1.0	-0.9757	0.9757	-0.0296	1.5866	2.1288	1.8593
2.0	-1.1179	1.1120	-0.0135	1.0782	2.2403	1.6486
3.0	-1.4854	1.4985	-0.0080	1.1568	2.5384	1.7099
4.0	-2.1041	2.1044	-0.0085	0.9603	2.6651	1.6614

Table 4.2: Instantaneous and mean values of C_L and C_D for R = 200, $A = 0.25$, $h = 1.25, \infty$ and $0.95 \leqslant f/f_0 \leqslant 4.0$.

with the case in which the free surface is absent indicates that the free surface presence at $h = 1.25$ does not seem to destabilize the periodicity of the lift and drag coefficients at low oscillation amplitude $A = 0.25$. This table also indicates that $\overline{C_L}$ values are affected significantly by the free surface presence at $h = 1.25$ for $0.95 \leqslant f/f_0 \leqslant 4.0$.

$\overline{C_L}$ takes on positive values in the interval $0.2561 \leqslant \overline{C_L} \leqslant 0.3926$ in the presence of a free surface located at $h = 1.25$; whereas the values of $\overline{C_L}$ are nearly zero at $h = \infty$, as expected. In contrast, inclusion of the free surface at a submergence depth of $h = 1.25$ has only a slight effect on $\overline{C_D}$ values for all frequency ratios. For example, for $f/f_0 = 1.0$ and $f/f_0 = 3.0$, the values of $\overline{C_D}$ increase by a factor of 1.03 and 1.11, respectively. The effect of increasing the frequency ratio, f/f_0, from 0.95 to 4.0 increases both $C_{L_{max}}$ and $C_{D_{max}}$ when $h = 1.25$. Table 4.2 indicates that the peak-to-peak amplitude of C_L at $h = 1.25$ tends to (i) increase slightly for $f/f_0 = 0.95$ and 1.0; (ii) decrease slightly for $f/f_0 = 2.0$ and 3.0 and (iii) increase slightly for $f/f_0 = 4.0$ when compared to $h = \infty$. On the other hand, the peak-to-peak amplitude of C_D at $h = 1.25$ tends to (i) increase slightly for $f/f_0 = 0.95$, 1.0 and 2.0 and (ii) decrease slightly for $f/f_0 = 3.0$ and 4.0 when compared to $h = \infty$.

4.2 Vortex formation modes and fluid forces: R = 200, $A = 0.5$ and $h = 1.25$

At $A = 0.5$, the effect of transverse oscillations become significant which leads to pronounced disturbance of the free surface and a changeover from one mode of vortex formation to another is observed as f/f_0 increases from 0.95 to 4.0. The free surface waves, in return, interact with the oscillating cylinder and have a great influence on the unsteady and mean fluid forces acting on the cylinder, mechanical energy transfer, as well as the formation of different vortex shedding modes in the near wake region. In the following subsections, the distinctive features of the fluid forces acting on the

cylinder, the mechanical energy transfer and associated vortex shedding modes are analyzed in detail.

4.2.1 Fourier analysis and Lissajous patterns of fluctuating fluid forces

Table 4.3 summarizes the effect of f/f_0 and h on $C_{L_{min}}$, $C_{L_{max}}$, $\overline{C_L}$; $C_{D_{min}}$, $C_{D_{max}}$ and $\overline{C_D}$. This table indicates that the presence of free surface at $h = 1.25$ causes significant increase in $\overline{C_L}$ values at each frequency ratio considered in $0.95 \leqslant f/f_0 \leqslant 4.0$ when compared to $h = \infty$. However, $\overline{C_L}$ takes on negative values near zero in the absence of a free surface, as expected. On the other hand, the free surface presence at a submergence depth of $h = 1.25$ has only a slight effect on $\overline{C_D}$ values for all frequency ratios. For example, for $f/f_0 = 1.0$ and $f/f_0 = 3.0$, the values of $\overline{C_D}$ increase by a factor of 1.04 and 1.11, respectively. The peak-to-peak amplitude of C_L at $h = 1.25$ tends to increase slightly for $0.95 \leqslant f/f_0 \leqslant 3.0$; whereas the peak-to-peak amplitude of C_L increases by a factor of 2.66 for $f/f_0 = 4.0$. The peak-to-peak amplitude of C_D at $h = 1.25$ increases slightly for frequency ratios $f/f = 0.95$ and 1.0. On the other hand, the peak-to-peak amplitude of C_D at $h = 1.25$ decreases slightly for $f/f \geqslant 2.0$. The effect of increasing the frequency ratio, f/f_0, from 0.95 to 4.0 increases the magnitude of $C_{L_{min}}$, $C_{L_{max}}$ and $C_{D_{max}}$ at $h = 1.25$.

Figure 4.4 shows the time history of the instantaneous lift coefficient, C_L, plotted with the displacement, $y(t)$, and the PSD of the lift coefficient. This is done by expanding C_L over ten periods of the cylinder oscillation, $10T$. When $f/f_0 = 0.95$

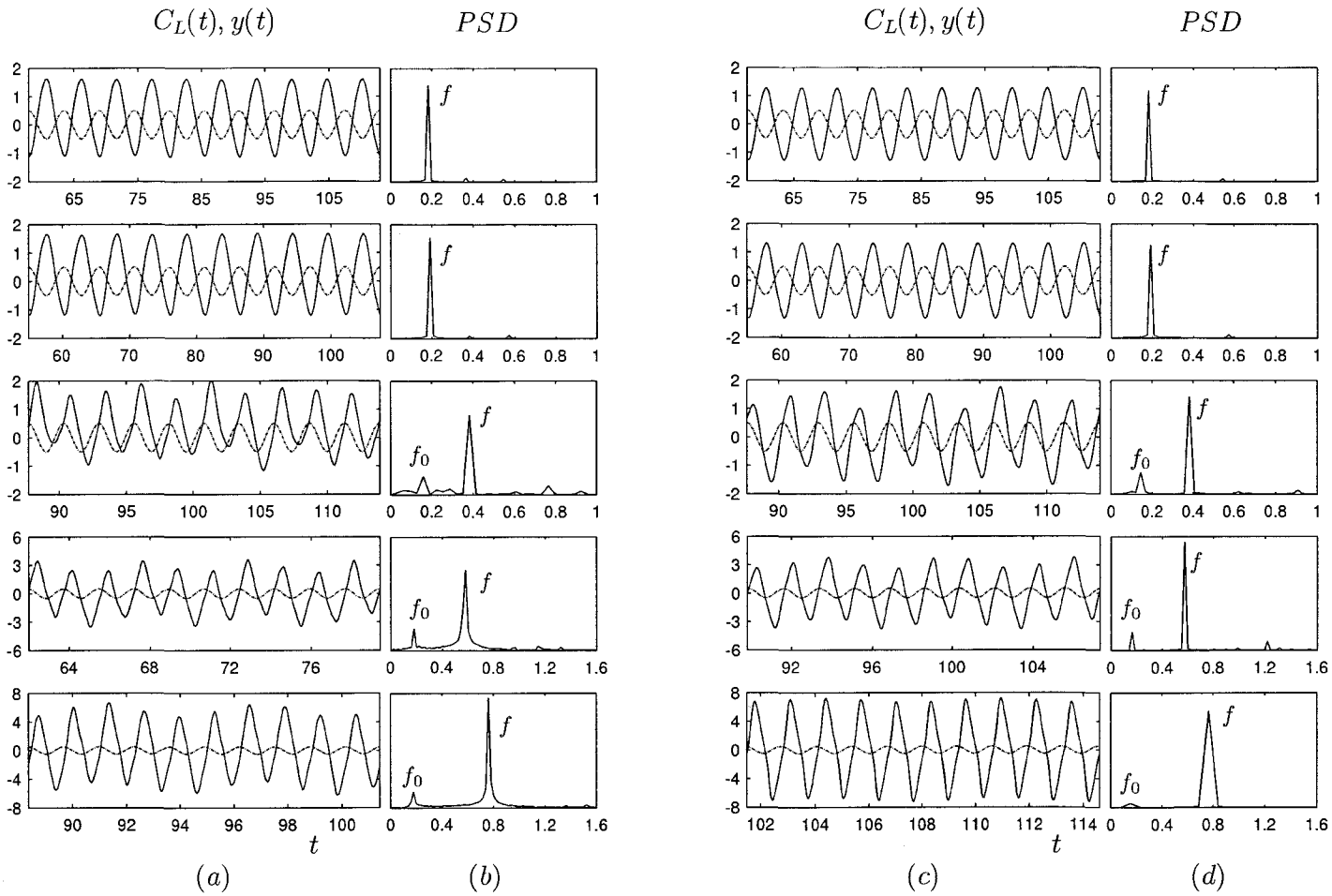

Figure 4.4: The time variation of the lift coefficient, C_L, —, and the transverse displacement, $y(t)$, $-\cdot-$, along with the PSD of C_L for R = 200, $A = 0.5$ and $f/f_0 = 0.95, 1.0, 2.0, 3.0, 4.0$ (from top to bottom): (a, b) $h = 1.25$ and (c, d) $h = \infty$.

$h = 1.25$						
f/f_0	$C_{L_{min}}$	$C_{L_{max}}$	$\overline{C_L}$	$C_{D_{min}}$	$C_{D_{max}}$	$\overline{C_D}$
0.95	-1.1439	1.6221	0.2899	1.5566	2.5958	2.0896
1.0	-1.2086	1.6830	0.2708	1.5734	2.8049	2.2106
2.0	-1.2086	2.0274	0.4051	1.0166	3.1711	2.1239
3.0	-3.3151	3.9645	0.2055	0.0330	4.8231	2.1772
4.0	-6.1159	6.5296	0.1117	-1.1249	6.0852	2.3165
$h = \infty$						
0.95	-1.2778	1.2779	-0.0020	1.5461	2.5251	2.0297
1.0	-1.3264	1.3216	-0.0048	1.5646	2.7080	2.1274
2.0	-1.7160	1.7620	-0.0042	1.0646	2.9911	1.9821
3.0	-4.1466	3.8162	-0.0389	-0.0493	4.2804	1.9555
4.0	-7.2159	7.2186	0.0014	0.3182	3.0298	1.6431

Table 4.3: Instantaneous and mean values of C_L and C_D for R = 200, $A = 0.5$, $h = 1.25, \infty$ and $0.95 \leqslant f/f_0 \leqslant 4.0$.

and $f/f_0 = 1.0$ the traces of C_L exhibit segments of repeatable signatures every cycle of cylinder oscillation, T, regardless of the free surface presence as shown in Figures 4.4(a) and 4.4(c). However, when the cylinder is submerged under the free surface the traces of C_L are seen to be less persistent. The PSD analysis of the signatures of C_L (see Figures 4.4(b) and 4.4(d)) shows that in all cases C_L fluctuates with a single frequency corresponding to the forcing frequency, f, of the cylinder oscillation.

Figure 4.5 shows the time history of instantaneous drag coefficient, C_D, plotted with the displacement, $y(t)$, and the PSD of the drag coefficient. This is done by expanding

C_D over six periods of cylinder oscillation, $6T$. The drag coefficient, C_D, shows highly periodic behaviour per $T/2$ when $h = \infty$ and per T when $h = 1.25$, at both $f/f_0 = 0.95$ and 1.0. This is also confirmed by the PSD analysis of C_D which indicates that C_D oscillates with a single frequency, $2f$, at $f/f_0 = 0.95$ and 1.0 when $h = \infty$. However, the PSD analysis of C_D predicts the existence of an additional frequency, f_0, when $h = 1.25$. Consequently, the C_D behaviour shows a competition between the dominant frequency, $2f$, and this additional frequency for $f/f_0 = 0.95$ and 1.0 when $h = 1.25$.

Lissajous patterns of C_L and C_D are shown in Figure 4.6. These plots are obtained by considering at least 10 cycles of the cylinder oscillation. Regardless of the free surface presence, the trajectories of $C_L(y)$ and $C_D(y)$ show a highly repetitive congruent form indicating phase-locking between the cylinder loading and its motion for $f/f_0 = 0.95$ and 1.0. This confirms our finding, based on the PSD analysis of C_L, that the time-dependent lift force, C_L, is phase-locked to the cylinder motion. The effect of the presence of free surface at $h = 1.25$ seems to cause a shift of the positive (clockwise) hysteresis loops of $C_D(y)$ in the left half-plane. In Figure 4.6 the direction of the hysteresis loops is indicated by arrows. The Lissajous patterns of $C_L(y)$ can be related to the mechanical energy transfer between the cylinder and the fluid for transverse oscillations of cylinder (see, for example, Blackburn and Henderson (1999)). Thus, for transversely oscillating cylinder, the amount of transfered mechanical energy can be then defined as the area enclosed by $C_L(y)$ in Figure 4.6(a) when $h = 1.25$ and Figure 4.6(c) when $h = \infty$. The Lissajous patterns of $C_L(y)$ enclose a larger area within the positive hysteresis loops when $h = \infty$ when compared to $h = 1.25$ for both $f/f_0 = 0.95$ and 1.0. This indicates that the presence of a free

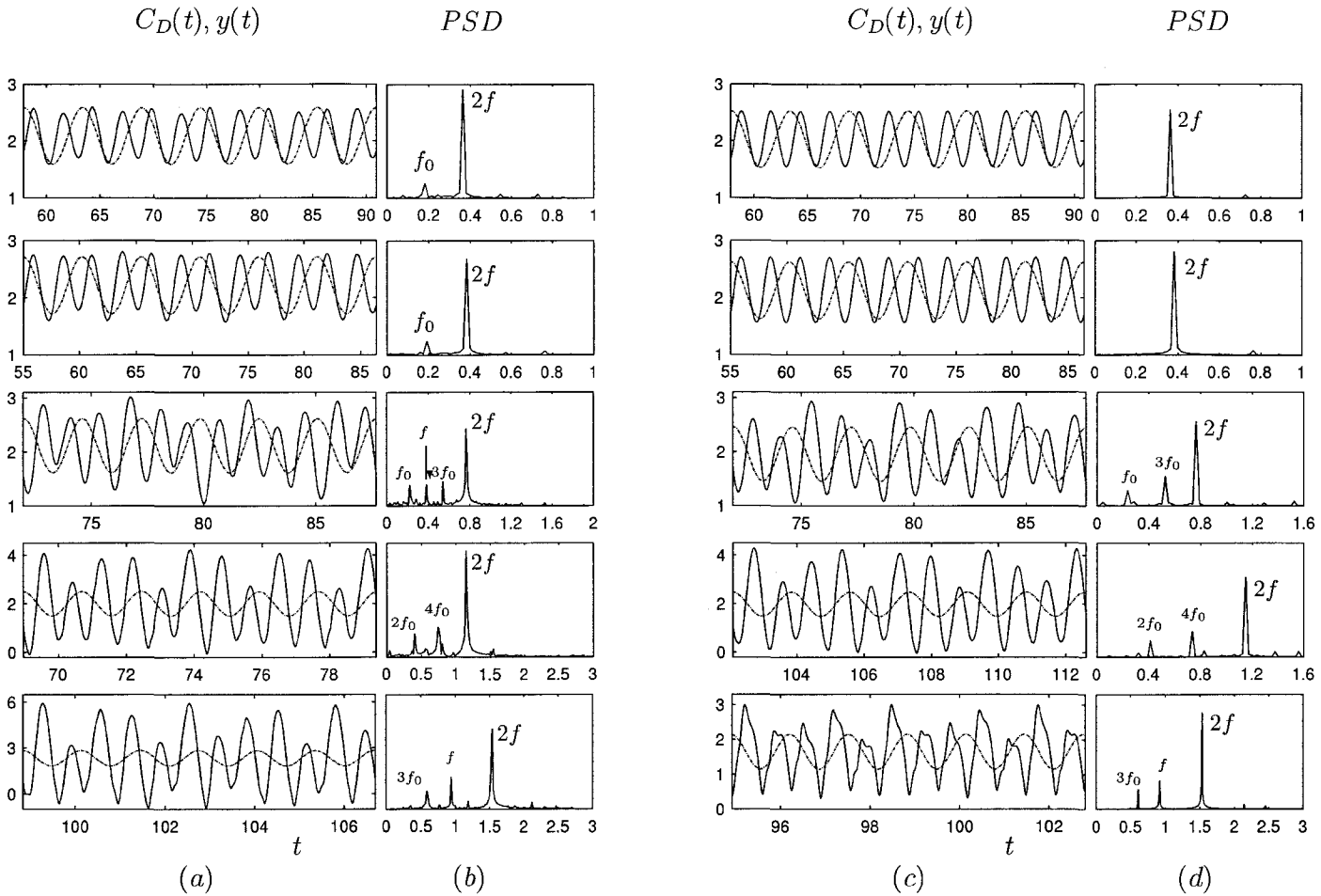

Figure 4.5: The time variation of the drag coefficient, C_D, —, and the transverse displacement, $y(t)$, $-\cdot-$, along with the PSD of C_D for R = 200, $A = 0.5$ and $f/f_0 = 0.95, 1.0, 2.0, 3.0, 4.0$ (from top to bottom): (a, b) $h = 1.25$ and (c, d) $h = \infty$.

surface at $h = 1.25$ induces the mechanical energy transfer between the fluid and the cylinder when $f/f_0 = 0.95$ and 1.0. It is noted that in the absence of a free surface the direction of the mechanical energy transfer is from the fluid to the cylinder when $f/f_0 = 0.95$ and 1.0.

At $f/f_0 = 2.0$, the presence of a free surface at $h = 1.25$ significantly alters C_L and C_D as shown in Figures 4.4 and 4.5. Figure 4.4(c) shows that the signatures of C_L exhibit almost a periodic trace over three cycles of the cylinder oscillation, $3T$, in the absence of a free surface. The presence of a free surface at $h = 1.25$ seems to shift the periodicity of C_L from $3T$ to $5T$ (see Figures 4.4(a) and 4.4(c)). Regardless of the free surface presence, the signatures of C_L are less persistent. This is consistent with the Lissajous patterns of $C_L(y)$ as shown in the third row of Figure 4.6. The PSD analysis of C_L predicts the existence of another frequency, f_0, in addition to the dominant frequency, f. The lift coefficient presented in Figure 4.4 reflects the combined effects of these frequencies on the time-dependent C_L signature. The end consequence of this is a loss of the persistent pattern of C_L.

The time-dependent signatures of drag coefficient, C_D, clearly show the almost periodic behaviour over $3T/2$ at $f/f_0 = 2.0$ when $h = \infty$ as displayed in Figure 4.5(c). The presence of a free surface at $h = 1.25$ seems to shift the periodicity of C_D from $3T/2$ to $2T$ (see Figures 4.5(a) and 4.5(c)). The PSD analysis of C_D exhibits two frequencies, f_0 and $3f_0$, in addition to the dominant frequency, $2f$, in the absence of a free surface (see Figure 4.5(d)). Interestingly, the additional frequency, f, is observed only in the presence of a free surface at $h = 1.25$ (see Figure 4.5(b)).

At $f/f_0 = 2.0$, the presence of a free surface at $h = 1.25$ seems to shift the positive

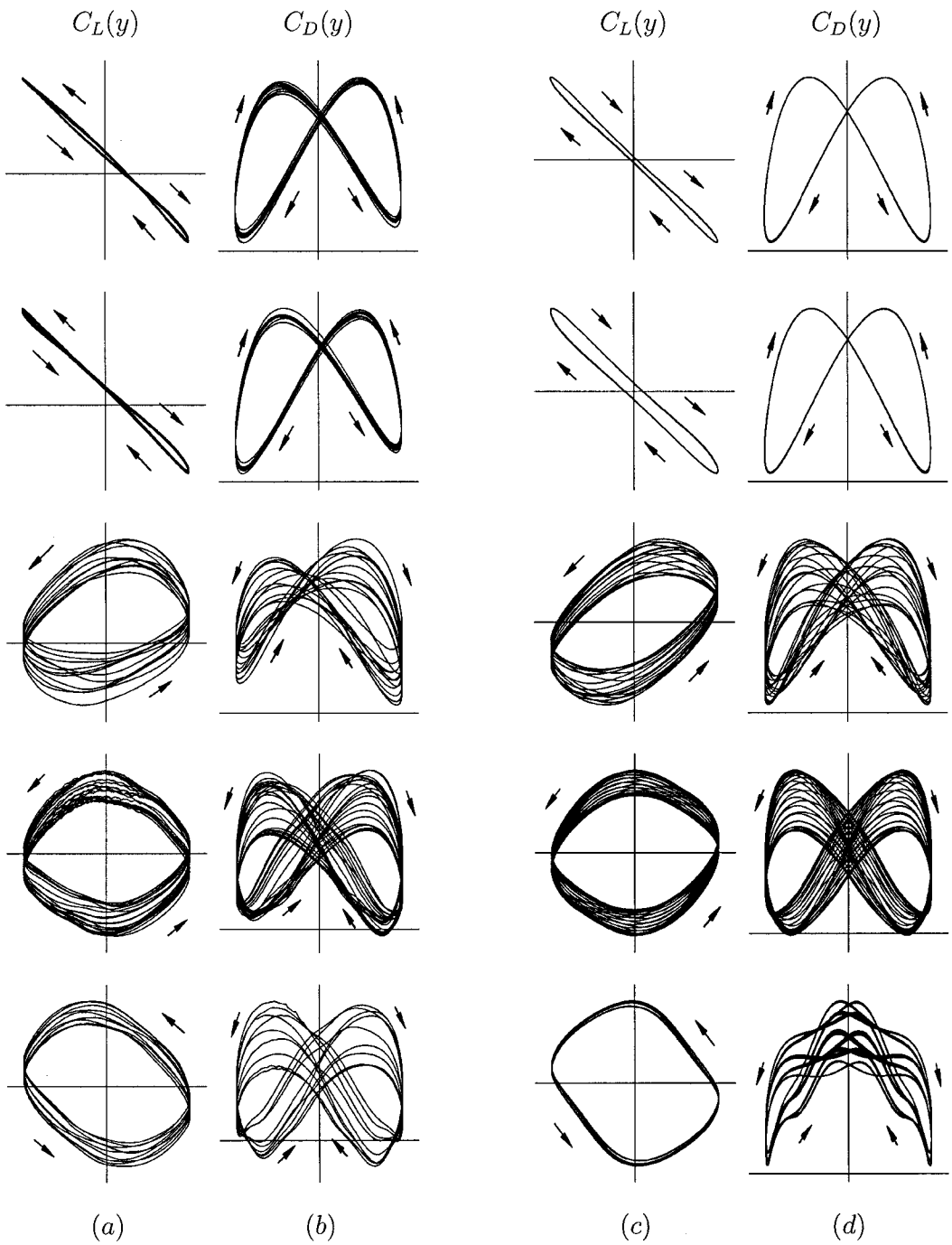

Figure 4.6: Lissajous patterns of C_L and C_D for R = 200, $A = 0.5$ and $f/f_0 = 0.95, 1.0, 2.0, 3.0, 4.0$ (from top to bottom): (a, b) $h = 1.25$ and (c, d) $h = \infty$.

hysteresis loops of $C_D(y)$ into the right half-plane. The trajectories of $C_D(y)$ are less congruent from one cylinder cycle to another, indicating an increase in phase variations between the cylinder motion and its loading. The corresponding Lissajous patterns of $C_L(y)$ in Figure 4.6 indicate a generally similar form regardless of the free surface presence. However, the area enclosed by $C_L(y)$ in Figure 4.6(a) is slightly smaller than that for $h = \infty$ (see Figure 4.6(c)). This observation emphasizes that the inhibiting influence of the free surface located at $h = 1.25$ is to reduce the mechanical energy transfer from the cylinder to the fluid. Figure 4.6 shows that the Lissajous patterns of $C_D(y)$ and $C_L(y)$ have well-defined trajectories but with a loss of phase-locking. The end consequence is that, at $f/f_0 = 2.0$, the cylinder loading is quasi-phase-locked to the cylinder motion regardless of the free surface presence.

At $f/f_0 = 3.0$, C_L exhibits a quasi-periodic behaviour, regardless of the free surface presence, as shown in Figure 4.4. C_L has a repeatable but not persistent pattern every three cycles of cylinder oscillation, $3T$, when $h = 1.25$ (see Figure 4.4(a)). On the other hand, this repeatability is observed every $4T$ in the absence of a free surface (see Figure 4.4(c)). Thus, the free surface presence at $h = 1.25$ seems to speed up the vortex shedding process, as will be confirmed in the next section. The PSD analysis of C_L predicts two well-defined frequencies, f and f_0, for both $h = 1.25$ and $h = \infty$.

The drag coefficient, C_D, exhibits a quasi-periodic behaviour per $3T/2$ when $h = \infty$ and per $3T$ when $h = 1.25$ at $f/f_0 = 3.0$. Regardless of the free surface presence, the traces of C_D are not persistent as shown in Figure 4.5. Table 4.3 shows that the presence of a free surface at $h = 1.25$ dramatically affects the value of $C_{D_{min}}$ at $f/f_0 = 3.0$ by shifting its value from -0.0493 to 0.0330. The PSD analysis of C_D

shows two frequencies, $2f_0$ and $4f_0$, in addition to the dominant frequency, $2f$, when $h = 1.25$ and $h = \infty$.

The Lissajous patterns of $C_D(y)$ and $C_L(y)$ show similar form regardless of the free surface presence as shown in the fourth row of Figure 4.6 when $f/f_0 = 3.0$. These patterns indicate a less congruent shape from one oscillation cycle to another. Consequently, in both cases there are increased phase variations between the loading and the cylinder motion, which emphasizes quasi-phase-locking. In Figures 4.6(a) and 4.6(b), at $h = 1.25$, the $C_L(y)$ and $C_D(y)$ hysteresis loops seem to show less repeatable patterns than that for $h = \infty$ (see Figures 4.6(c) and 4.6(d)). Thus, the free surface presence at $h = 1.25$ seems to destabilize the fluctuations of the C_L and C_D relative to the cylinder displacement, $y(t)$. As for the mechanical energy transfer, the presence of a free surface causes mechanical energy transfer loss from the cylinder to the fluid.

For $f/f_0 = 4.0$, the trace of C_L tends to have a fully periodic behaviour per $5T$ in the absence of a free surface. Free surface presence at $h = 1.25$ seems to shift periodicity of C_L per $5T$ to quasi-periodicity per $4T$ (see Figures 4.4(a) and 4.4(c)). The PSD analysis of C_L exhibits the existence of another frequency, f_0, in addition to the dominant frequency, f. The lift coefficient presented in Figure 4.4(a) reflects the combined effects of these frequencies on the time-dependent C_L signature when $h = 1.25$. This results in a C_L pattern which is less persistent at $h = 1.25$.

The time-dependent signatures of the drag coefficient, C_D, exhibit quasi-periodic behaviour over $5T/2$ when $f/f_0 = 4.0$ and $h = \infty$ as shown in Figure 4.5(c). The presence of a free surface at $h = 1.25$ seems to shift the quasi-periodicity of C_D from

$5T/2$ to $3T$. Regardless of the free surface presence, the signatures of C_D are not persistent. The PSD analysis of C_D shows two frequencies, f and $3f_0$, in addition to the dominant frequency, $2f$, when $h = 1.25$ and $h = \infty$ (see Figures 4.5(b) and 4.5(d)). Table 4.3 shows that the presence of a free surface at $h = 1.25$ dramatically affects the value of $C_{D_{min}}$ at $f/f_0 = 4.0$ by shifting its value from 0.3182 to -1.1249.

The presence of a free surface at $h = 1.25$ seems to shift the negative hysteresis loops of $C_D(y)$ into the left half-plane. The Lissajous patterns of $C_D(y)$ at $h = \infty$ indicate a complex form compared with those at $h = 1.25$ (see Figures 4.6(b) and 4.6(d)). Regardless of the free surface presence, the Lissajous patterns of $C_D(y)$ show non-congruent form, indicating a loss of phase-locking between the cylinder motion and its loading. Thus, at $f/f_0 = 4.0$, $C_D(y)$ is quasi-phase-locked to the cylinder motion when $h = \infty$ and $h = 1.25$. Figures 4.6(a) and 4.6(c) show that the Lissajous patterns of $C_L(y)$ are more well defined and congruent at $h = \infty$ than those at $h = 1.25$. The end consequence of all these is that, at $f/f_0 = 4.0$, there is a transition from the phase-locking of C_L at $h = \infty$ to the quasi-phase-locking of C_L at $h = 1.25$. The area enclosed by $C_L(y)$ in Figure 4.6(a) is slightly smaller than that at $h = \infty$ (see Figure 4.6(c)). This observation emphasizes that the presence of the free surface located at $h = 1.25$ reduces the mechanical energy transfer from the cylinder to the fluid.

4.2.2 Vortex formation modes in the near wake region

Figure 4.7 shows the equivorticity patterns in the near wake region for the frequency ratios $f/f_0 = 0.95$ and 1.0. At both of these frequencies the vortex shedding mode is

the locked-on asymmetric **2S** mode, per T, in which two vortices are alternatively shed from the upper and lower sides of the cylinder over one period of cylinder oscillations for both $h = 1.25$ and $h = \infty$. The free surface presence at $h = 1.25$ seems to increase the vortex formation length slightly ($\approx 11\%$). In addition, the shed vortex becomes weaker as it moves downstream due to its interaction with the free surface interface. The negative vorticity reaches its maximum amount at the instants $t = 0T$ and T, whereas the positive maximal vorticity amount is observed at $t = T/2$. When the fully developed negative vortex propagates from the cylinder ($t \approx T/2$), a positive peak occurs on the C_L trace. Similarly, the development and propagation of the positive vortex produces a negative peak on the C_L trace at the instants $t = 0T$ and T.

For frequency ratios $f/f_0 \geqslant 2.0$, the vortex shedding process becomes more complicated due to the strong interaction between the body of the cylinder and the shed vortices, the free surface interface and the shed vortices. The immediate coalescence of co-rotating vortices is observed in the near wake region for frequency ratios $f/f_0 \geqslant 2.0$ regardless of the free surface presence. It is noted that Flemming and Williamson (2005) reported a similar phenomena in their experimental work for combined transverse and in-line motion of a pivoting cylinder at R ≈ 700. The equivorticity plots over $5T$ are shown in Figure 4.8 for $f/f_0 = 2.0$ when $h = 1.25$ and $h = \infty$. In the absence of a free surface, a pair of positive co-rotating vortices develops from the upper side of the cylinder, over $3T/2$, and then coalesce immediately after the shedding of the second vortex ($3T/2 \leqslant t \leqslant 2T$) to form a single large vortex at the instant $t \approx 2T$, as shown in Figure 4.8(b). After this time period, a pair of negative co-rotating vortices is formed from the lower side of the cylinder which causes a fully developed positive vortex to shed from the upper side of the cylinder at $t \approx 3T$. The

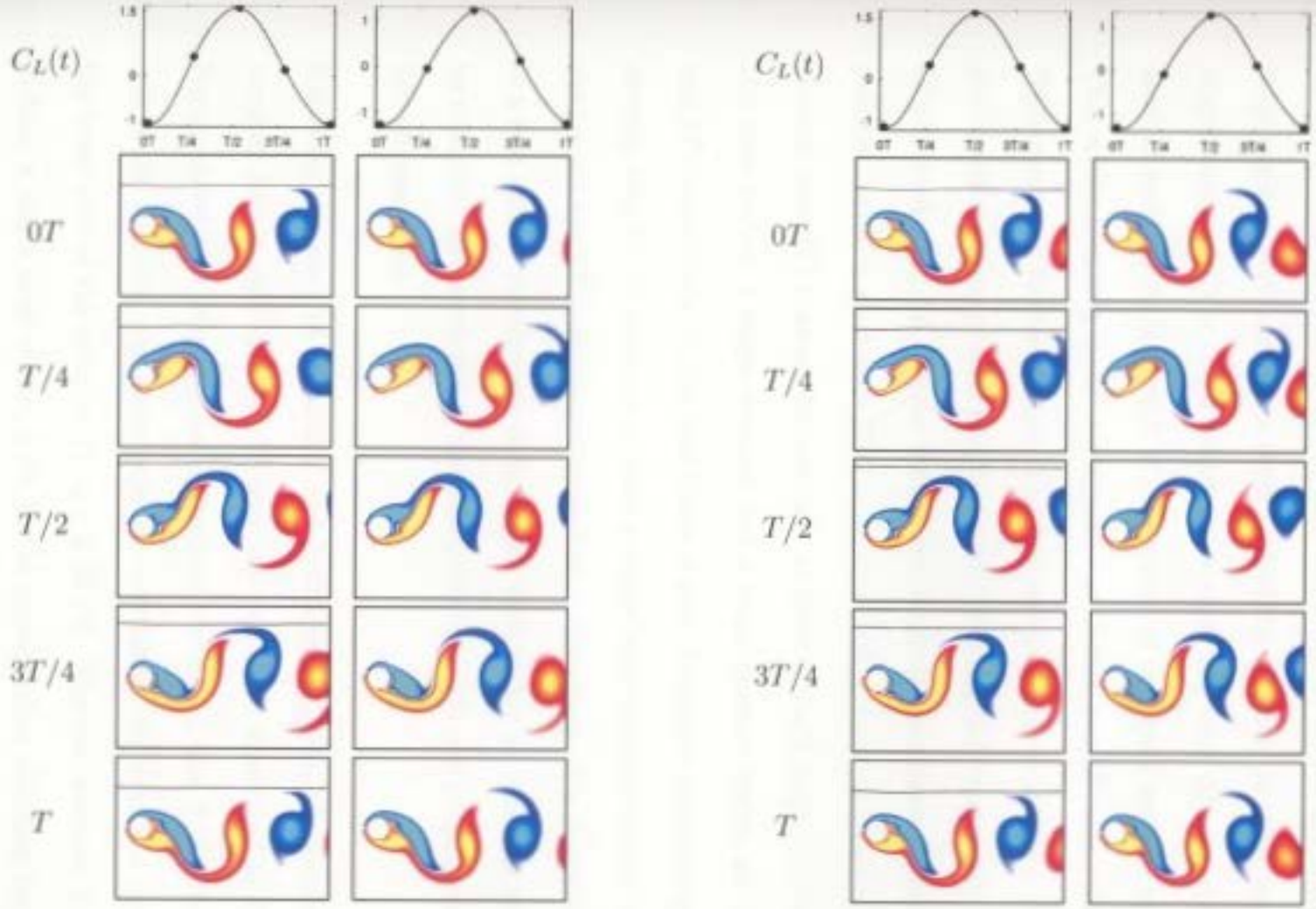

Figure 4.7: Equivorticity patterns over one period of cylinder oscillation, T, for R = 200, $A = 0.5$ and (a, b): $f/f_0 = 0.95$ ($T \approx 5.51$, $113 \leqslant t \leqslant 118.4$); (c, d): $f/f_0 = 1.0$ ($T \approx 5.24$, $112.6 \leqslant t \leqslant 117.8$). (a, c): $h = 1.25$; (b, d): $h = \infty$. The instantaneous snapshots are marked in the corresponding lift coefficient curves shown above the snapshots.

inverse process happens over the next $3T/2$, which causes the shedding of a single large negative vortex from the lower part of the cylinder. In other words, the cylinder alternatively sheds a single vortex from each side over $3T$, in which the development of each vortex is a result of coalescence of two weak vortices in the shear layer. This vortex shedding mode is designated as **C(2S)**, per $3T$, using the terminology of Williamson and Roshko (1988). The free surface presence at $h = 1.25$ seems to delay the vortex shedding process and the vortex shedding is characterized by the locked-on **C(4S)** mode, per $5T$. In the first phase of the development of this mode, a single positive vortex is shed over $3T/2$, and a pair of positive co-rotating vortices (which develop over $2T$) coalesce immediately to form a single large positive vortex. After this time period, a single negative and a single positive vortex are shed, over $5T/2$ and $3T$, respectively. In the final phase, a pair of negative co-rotating vortices (which develop over $9T/2$) coalesce to form a single large negative vortex which is shed at the instant $t = 5T$ as shown in Figure 4.8(a). This figure also shows that the presence of a free surface significantly deforms the shed vortex so that this vortex appears to be elongated and weakened in the transverse direction and propagates slowly into the downstream wake.

Following the work of Flemming and Williamson (2005), the mechanism of vortex merging for the case of $h = 1.25$ over two periods is illustrated in Figure 4.9. This Figure shows that two positive co-rotating vortices, 1 and 2, develop on the upper side of the cylinder, and three negative co-rotating vortices, 3, 4 and 5, develop on the lower side of the cylinder ($T \leqslant t \leqslant 3T/2$). The two vortices, 1 and 2, coalesce to form a single large vortex, 1+2, in the upper vortex shedding layer ($t = 2T$). In addition, two of the three negative vortices, 3 and 4, coalesce to form the vortex 3+4

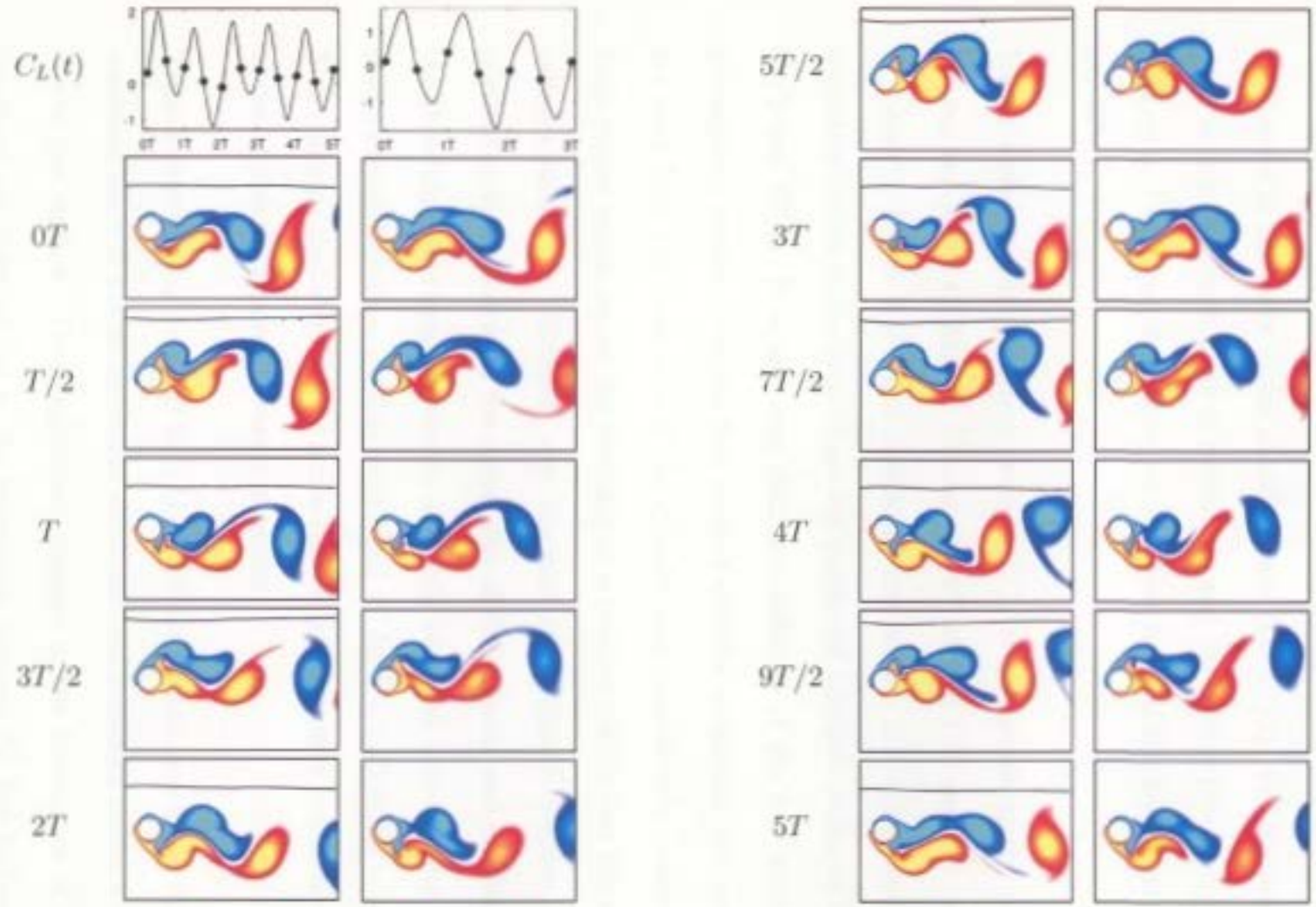

Figure 4.8: Equivorticity patterns over five periods of cylinder oscillation, $5T$, for R = 200, $A = 0.5$ and $f/f_0 = 2.0$ ($T \approx 2.62$, $100.6 \leqslant t \leqslant 113.8$): (a) $h = 1.25$ and (b) $h = \infty$. The instantaneous snapshots are marked in the corresponding lift coefficient curves shown above the snapshots.

while the vortex 5 remains detached. Thus, a single large vortex involving 3+4 and 5 is formed in the lower vortex shedding layer ($t = 3T/2$). The whole process repeats itself within the fourth cycle of oscillation, and the reverse process happens over the fifth cycle. Consequently, the vortex shedding mode is the asymmetric **C(4S)** mode per $5T$.

Futher increasing the frequency ratio to $f/f_0 = 3.0$, speeds up the vortex shedding process and yields **C(2S)** per $3T$ and per $4T$ for $h = 1.25$ and $h = \infty$, respectively, as shown in Figure 4.10. The significant free surface distortion produces a notable secondary vortex in the area where the maximum inflation occurs at $t = T/2$, $3T/2$, $5T/2$ and $7T/2$. It is also seen that this inflation of the free surface rounds the propagating vortex. Over the first cycle of cylinder oscillation, two negative vortices are shed from the lower side of the cylinder and immediately coalesce to form one large vortex which causes the shedding of a positive vortex over the second cycle of cylinder oscillation. Similarly, over the next two cycles of cylinder oscillation, two positive vortices shed from the upper side of the cylinder and coalesce which causes the shedding of the negative vortex. It should be noted that the near wake structure shows a quasi-locked-on pattern per $3T$ and $4T$ for $h = 1.25$ and $h = \infty$, respectively. This confirms that the effect of a free surface presence at $h = 1.25$ and $f/f_0 = 3.0$ is to destabilize the process of vortex shedding.

Typical equivorticity patterns for $f/f_0 = 4.0$ are plotted over $5T$ in Figure 4.11. The combined effect of a large oscillation amplitude and frequency causes a large distortion of the free surface. Thus, significant changes in the kinematics of the near wake structure are observed due to the interaction between the free surface vorticity and

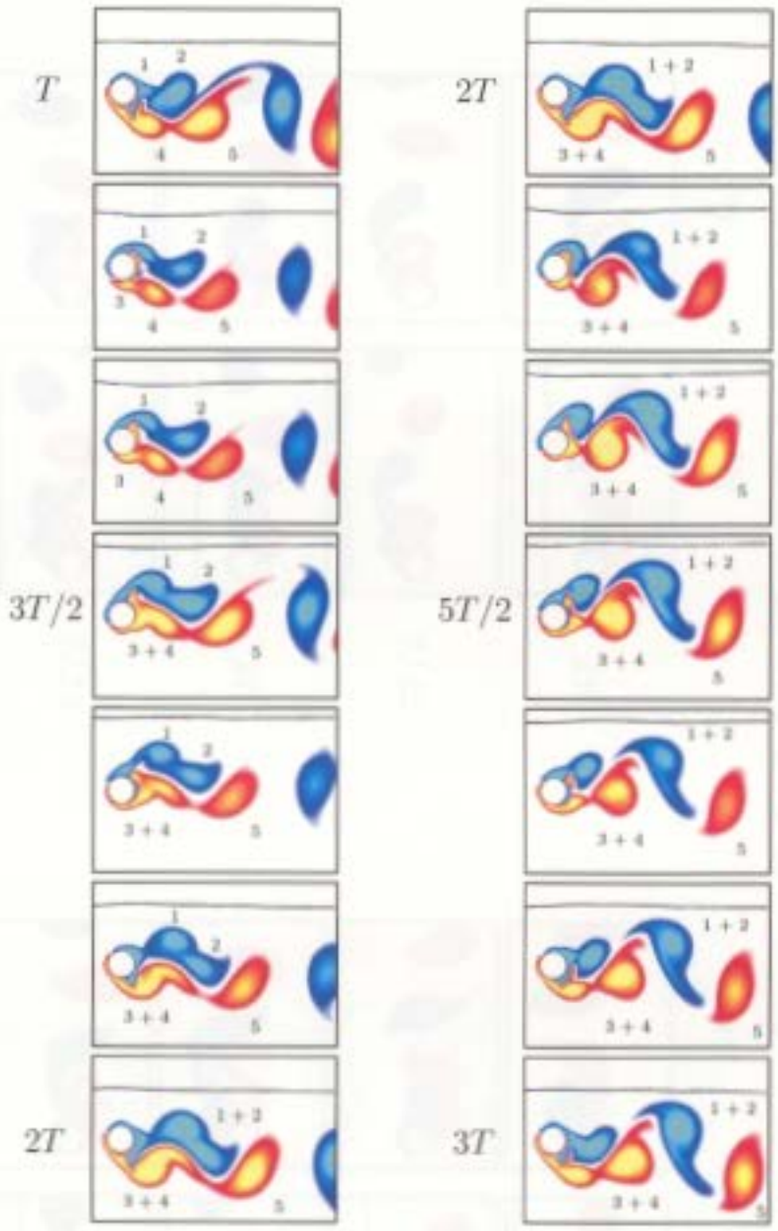

Figure 4.9: Mechanism of vortex coalescence in the near wake region for R = 200, $A = 0.5$, $h = 1.25$ and $f/f_0 = 2.0$ ($T \approx 2.62$, $103.2 \leqslant t \leqslant 108.4$).

the vorticity generated by the cylinder motion. The vortex shedding process produces the asymmetric **C(2S)** mode, per $5T$, for $h = \infty$ and the **C(4S)** mode, per $4T$, for $h = 1.25$, as shown in Figure 4.11. In the absence of the free surface, Figure 4.11(b) shows a complex asymmetric pattern in which two co-rotating negative vortices shed

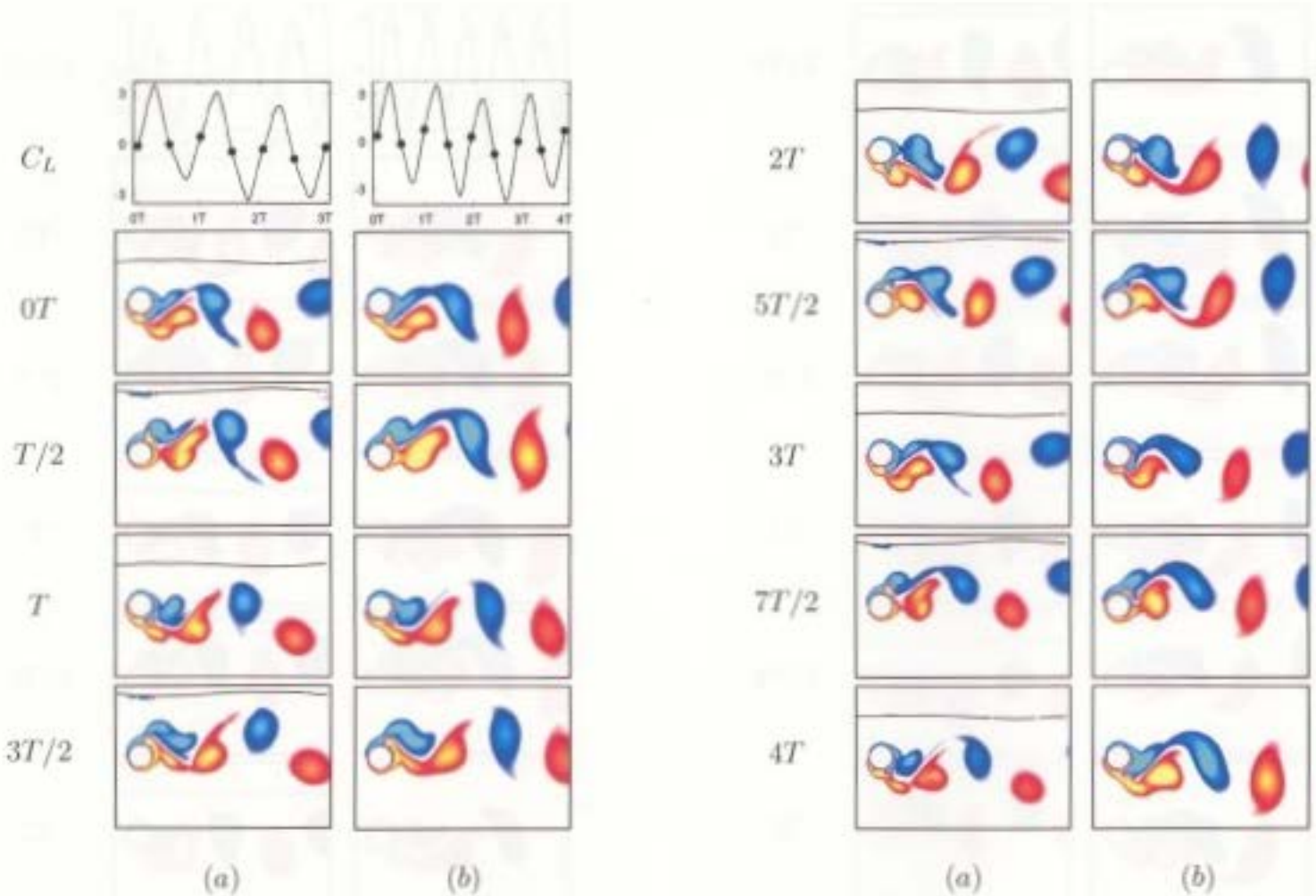

Figure 4.10: Equivorticity patterns over four periods of cylinder oscillation, $4T$, for R = 200, $A = 0.5$ and $f/f_0 = 3.0$ ($T \approx 1.75$, $77.66 \leqslant t \leqslant 82.9$): (a) $h = 1.25$ and (b) $h = \infty$. The instantaneous snapshots are marked in the corresponding lift coefficient curves shown above the snapshots.

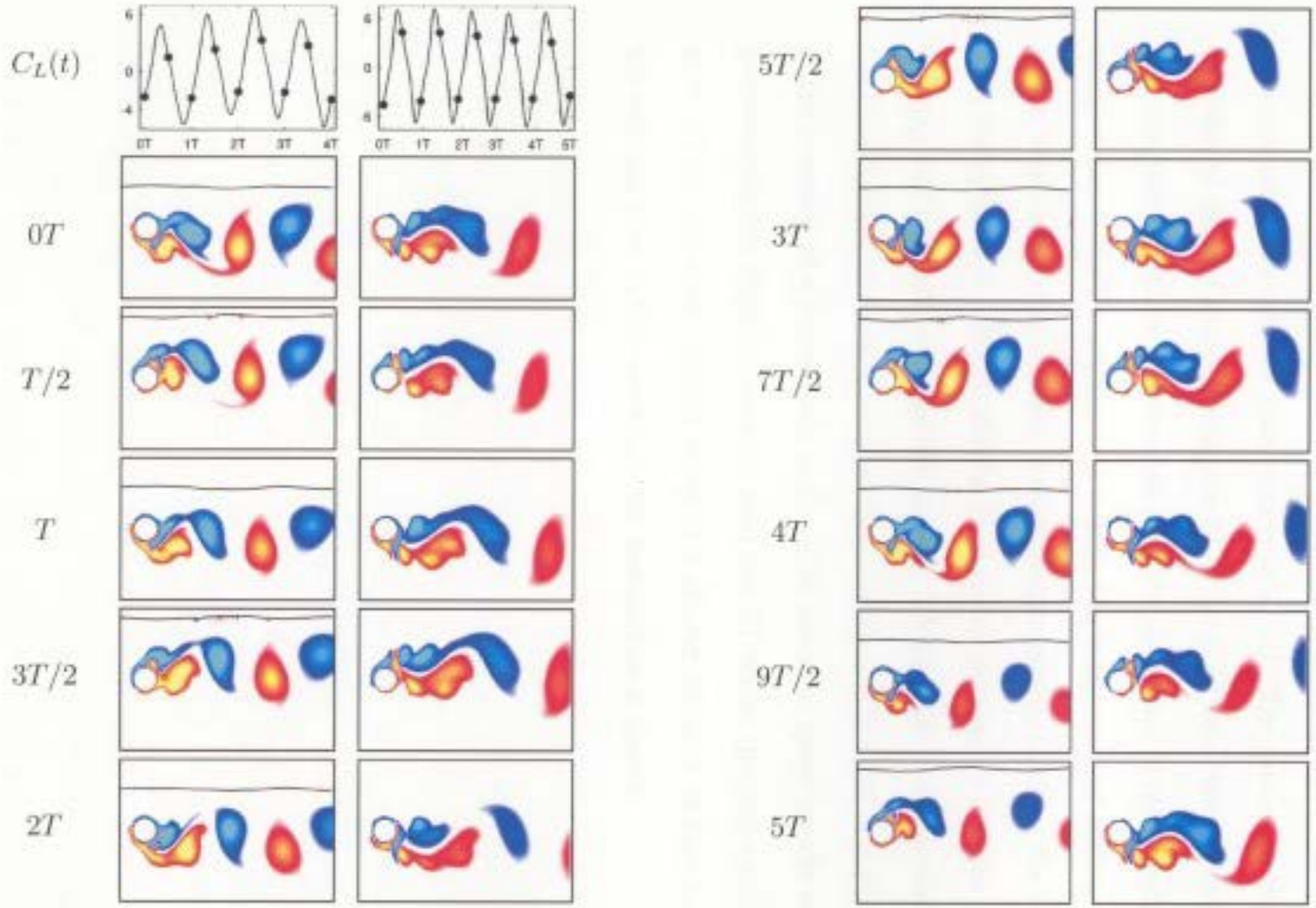

Figure 4.11: Equivorticity patterns over five periods of cylinder oscillation, $5T$, for R = 200, $A = 0.5$ and $f/f_0 = 4.0$ ($T \approx 1.3$, $88.3 \leqslant t \leqslant 93.5$): (a) $h = 1.25$ and (b) $h = \infty$. The instantaneous snapshots are marked in the corresponding lift coefficient curves shown above the snapshots.

from the lower side of the cylinder coalesce to form a single vortex at a distance of approximately five cylinder diameters ($t = 5T/2$). The inverse process leads to the shedding of another single vortex over $5T/2 \leqslant t \leqslant 5T$ as a result of the coalescence of two co-rotating positive vortices shed from the upper side of the cylinder. However, in the presence of the free surface at $h = 1.25$, the coalescence of vortices appears in the region very close to the cylinder surface. Consequently, tracking the vortex shedding from the cylinder is difficult. As the shed vortex propagates into the downstream, it enlarges and interacts with the free surface. This results in free surface inflation and the development of secondary vorticity ($t = T/2$, $3T/2$, $5T/2$ and $7T/2$). The effect of the presence of a free surface at $h = 1.25$ seems to speed up the vortex shedding process since two single vortices are shed over $4T$ unlike the case when $h = \infty$. Thus, at $h = 1.25$, the weak vortices decay at a shorter distance as they move away from the cylinder than in the case where the free surface is absent.

Chapter 5

Summary, conclusions and recommendations for future work

The Reynolds transport theorem is extended for control volumes containing a fluid interface for deriving the governing equations of fluid mechanics for the first time. This is done by utilizing, mainly, generalized differentiation. The use of generalized function theory made this derivation straightforward and systematic. The resulting integral form of the governing equations is used to design a numerical simulation tool within an object-oriented software framework for solution of a class of free surface problems with arbitrarily moving rigid bodies.

The fluid flow governed by the full Navier-Stokes equations is only modeled within a two-dimensional computational domain. The motion of the air is neglected and the effect of the ambient pressure exerted on the fluid by the air is taken into consider-

ation to state the free surface boundary conditions. The no-slip boundary condition is implemented on the rigid body. The method of solution is based on a finite volume discretization of the unsteady Navier-Stokes equations in their pressure-velocity formulation on a fixed Cartesian grid. The combined volume of fluid and fractional area/volume obstacle representation method, and the cut cell method are employed to track the fluid-air and fluid-body interfaces. Well-posed open boundary conditions are applied at the outflow part of the computational domain. Discrete pressure and velocity fields corresponding to successive time instants are obtained as a result of solving a coupled sparse linear system in primitive variables. The numerical model is implemented in a computer program, written in the C++ programming language, using object-oriented, data abstraction and template metaprogramming paradigms. The use of a high level programming language like C++ allows us to define concepts specific to the CFD application domain directly in the source code. The CFD code is represented as a set of objects which interact with each other to achieve the desired functionality. A program designed in this way turns out to be less complex and thus, is much easier to understand and maintain. The size of the manually written code is reduced up to four times without loss of run-time efficiency by automatically generating the code corresponding to both Cartesian axes from a single C++ template augmented with compile-time optimizations and function inlining.

This numerical simulation tool is applied to the problem of unsteady, laminar, two-dimensional flow of a viscous incompressible fluid past a transversely oscillating circular cylinder in the presence of a free surface. The numerical simulations are conducted at a Reynolds number of R = 200 for a fixed submergence depth, $h = 1.25$, and displacement amplitudes $A = 0.25$ and 0.5 in the frequency ratio range

$0.95 \leqslant f/f_0 \leqslant 4.0$. The results are analyzed by means of equivorticity patterns in the near wake region as well as the fluid forces acting on the cylinder. The main objective is to address the alterations of loading on cylinder and locked-on vortex formation modes due to the presence of a free surface. The PSD analysis of the lift coefficient, Lissajous representations of the lift coefficient and equivorticity patterns in the near wake are used to determine locked-on vortex shedding modes. The Lissajous representations of the lift and drag coefficients are also used to demonstrate the mechanical energy transfer, degree of phase-locking or a loss of lock-on and associated phase shift. The code validations are presented in the special cases of uniform flow past (i) a stationary cylinder (no forced oscillations); (ii) a cylinder undergoing forced rotational oscillations; (iii) a cylinder undergoing forced transverse oscillations in the absence of a free surface, (iv) uniform flow past a stationary cylinder in the presence of a free surface. Good comparisons with previous experimental and numerical results are obtained.

5.1 Effect of the free surface at $h = 1.25$ and the frequency ratio, f/f_0, on the vortex formation modes and fluid forces: R = 200, $A = 0.25, 0.5$

Table 5.1 summarizes the combined effect of the free surface presence at $h = 1.25$ and f/f_0 on the vortex shedding modes, fluctuating fluid forces, C_L and C_D, and their periods. This table also displays the results obtained in the absence of a free surface

($h = \infty$) to determine the effect of the free surface at $h = 1.25$ on the shedding modes as well as C_L and C_D. For low frequency ratios, $f/f_0 = 0.95$ and 1.0, the classical

$h = 1.25$					
f/f_0	Mode	Period of C_L and modes	Period of C_D	C_L and $y(t)$ phase locking	C_D and $y(t)$ phase locking
0.95	**2S**	1T	1T	locked	locked
1.0	**2S**	1T	1T	locked	locked
2.0	**C(4S)**	5T	2T	quasi-locked	quasi-locked
3.0	**C(2S)**	3T	3T	quasi-locked	quasi-locked
4.0	**C(4S)**	4T	3T	quasi-locked	quasi-locked
$h = \infty$					
0.95	**2S**	1T	T/2	locked	locked
1.0	**2S**	1T	T/2	locked	locked
2.0	**C(2S)**	3T	3T/2	quasi-locked	quasi-locked
3.0	**C(2S)**	4T	3T/2	quasi-locked	quasi-locked
4.0	**C(2S)**	5T	5T/2	locked	quasi-locked

Table 5.1: The effects of the free surface at $h = 1.25$ and frequency ratio, $0.95 \leqslant f/f_0 \leqslant 4.0$, on the vortex shedding modes, C_L and C_D for R $= 200$ and $A = 0.5$.

Kármán vortex pattern (**2S** mode) occurs per one cycle of cylinder oscillation, T, regardless of the free surface presence, i.e., for both $h = 1.25$ and $h = \infty$. The corresponding PSD of each C_L confirms that the vortex shedding mode period is T at these low frequency ratios when $h = 1.25$ and $h = \infty$. It is interesting to note that the effect of a free surface at $h = 1.25$ seems to be a period doubling in C_D for the two frequency ratios $f/f_0 = 0.95$ and 1.0. As a result of increasing f/f_0 from 1.0 to 4.0 in

the presence of a free surface the near wake vorticity breaks up to produce different modes of vortex shedding: **2S** per T, **C(4S)** per $5T$, **C(2S)** per $3T$ and **C(4S)** per $4T$ when $h = 1.25$. However, the common vortex shedding mode **C(2S)** is obtained for the frequency ratios $f/f_0 \geqslant 2.0$ in the absence of the free surface ($h = \infty$). Thus, in the absence of a free surface, the vortex shedding modes are **2S**, per T, for $f/f_0 = 0.95$ and 1.0, and **C(2S)** per $3T$, $4T$ and $5T$, for $f/f_0 \geqslant 2.0$. This is consistent with the experimental findings of Williamson and Roshko at R = 200 (see Figure 4.1). Table 5.1 also shows that, at $f/f_0 = 0.95$ and 1.0, C_L and C_D are phase-locked to the cylinder displacement, $y(t)$, regardless of the free surface presence. In contrast, at $f/f_0 = 2.0$ and 3.0, C_L and C_D are quasi-phase-locked to $y(t)$ for both $h = 1.25$ and $h = \infty$. Interestingly, in the absence of a free surface at the high frequency ratio, $f/f_0 = 4.0$, there is a switchover from phase-locked C_L to quasi-phase-locked C_L when $h = 1.25$. At $f/f_0 = 4.0$, C_D is quasi-locked for both $h = 1.25$ and $h = \infty$.

The effect of increasing the oscillation frequency ratio beyond $f/f_0 = 1.0$ seems to increase the value of the period of the vortex shedding modes from T to $5T$ when $h = \infty$. Thus, in the presence of a free surface at $h = 1.25$, a changeover from one mode of vortex formation to another one is observed with the increase of f/f_0 beyond 1.0. In all cases considered C_L exhibits a periodic behaviour which is consistent with the vortex shedding modes and their periods. This confirms our findings that vortex lock-on occurs in these cases. However, a changeover in the nature of the fluctuations of the drag coefficient is observed with an increase in the frequency ratio.

In the fluid-cylinder interactions, the effect of the vortex shedding is to prompt a mechanical energy transfer between the cylinder and the fluid. The sign (direction

of hysteresis loops of C_L) of the energy transfer determines whether transverse oscillations of the cylinder will tend to increase or decrease in amplitude of the cylinder motion and, in many cases, knowledge of the mechanism of mechanical energy transfer is critical to design considerations. A signicant feature of the transverse oscillations of the cylinder is that the vortex formation and shedding are very sensitive to the frequency of the cylinder oscillation. This also affects the mechanism of mechanical energy transfer. The Lissajous patterns of C_L in Figure 4.6 show that when $A = 0.5$ at $f/f_0 = 0.95$ and 1.0 the presence of a free surface at $h = 1.25$ reduces the amount of mechanical energy transfered from the fluid to the cylinder. It is particularly interesting to note that the presence of free surface also induces the transfer of mechanical energy between the cylinder and the fluid. For frequency ratios $2.0 \leqslant f/f_0 \leqslant 4.0$ the effect of increasing f/f_0 is to enlarge the amount of energy transfered from the cylinder to the fluid regardless of the free surface presence. However, a comparison of the areas enclosed by C_L in Figure 4.6(a) with those given in Figure 4.6(c) indicates that the presence of the free surface at $h = 1.25$ reduces the amount of transferred mechanical energy.

Table 5.2 shows the combined effect of the free surface presence at $h = 1.25$ and an increase of the transverse amplitude, A, from 0.25 to 0.5, on the mechanism of mechanical energy transfer. In this table '$\leftrightarrow$' indicates a mechanical energy transfer between the cylinder and the fluid, '$\leftarrow$' indicates a mechanical energy transfer from the fluid to the cylinder and '$\rightarrow$' indicates a mechanical energy transfer from the cylinder to the fluid. It is clearly seen that regardless of the value of the transverse amplitude, the mechanical energy is always transfered from the fluid to the cylinder for $f/f_0 = 0.95$ and 1.0 and from the cylinder to the fluid for $f/f_0 \geqslant 2$ when $h = \infty$.

However, the presence of a free surface located at $h = 1.25$ dramatically alters the mechanism of mechanical energy transfer when $A = 0.5$, $f/f_0 = 0.95$ and 1.0 by inducing an energy transfer between the cylinder and the fluid.

f/f_0	$A = 0.25$		$A = 0.5$	
	$h = 1.25$	$h = \infty$	$h = 1.25$	$h = \infty$
0.95	cylinder $\leftarrow$ fluid	cylinder $\leftarrow$ fluid	cylinder $\leftrightarrow$ fluid	cylinder $\leftarrow$ fluid
1.0	cylinder $\leftarrow$ fluid	cylinder $\leftarrow$ fluid	cylinder $\leftrightarrow$ fluid	cylinder $\leftarrow$ fluid
2.0	cylinder $\rightarrow$ fluid	cylinder $\rightarrow$ fluid	cylinder $\rightarrow$ fluid	cylinder $\rightarrow$ fluid
3.0	cylinder $\rightarrow$ fluid	cylinder $\rightarrow$ fluid	cylinder $\rightarrow$ fluid	cylinder $\rightarrow$ fluid
4.0	cylinder $\rightarrow$ fluid	cylinder $\rightarrow$ fluid	cylinder $\rightarrow$ fluid	cylinder $\rightarrow$ fluid

Table 5.2: The effect of the transverse amplitude, $A = 0.25, 0.5$, on the mechanical energy transfer between the cylinder and the fluid for R $= 200$, $h = 1.25, \infty$ and $0.95 \leqslant f/f_0 \leqslant 4.0$.

Figure 5.1 summarizes the effects of increasing the frequency ratio, f/f_0, on the equivorticity patterns in the case when $h = 1.25$. At low transverse amplitude, $A = 0.25$, there is a substantial decrease in the vortex formation length as the frequency ratio f/f_0 increases from 0.95 to 4.0. The typical vortex shedding mode is the **2S** mode for all frequency ratios. At $A = 0.5$, the **2S** mode persists for $f/f_0 = 0.95$ and 1.0. The effect of increasing f/f_0 from 2.0 to 4.0 is to disturb the free surface significantly which, in turn, produces secondary vorticity yielding complex vortex formation patterns as shown in Figure 5.1(b). For $f/f_0 \geqslant 2.0$ the typical vortex shedding modes are **C(4S)** per $5T$, **C(2S)** per $3T$ and **C(4S)** per $4T$ when $h = 1.25$. For frequency ratios beyond 1.0, the vortex formation process seems to speed up at both $A = 0.25$

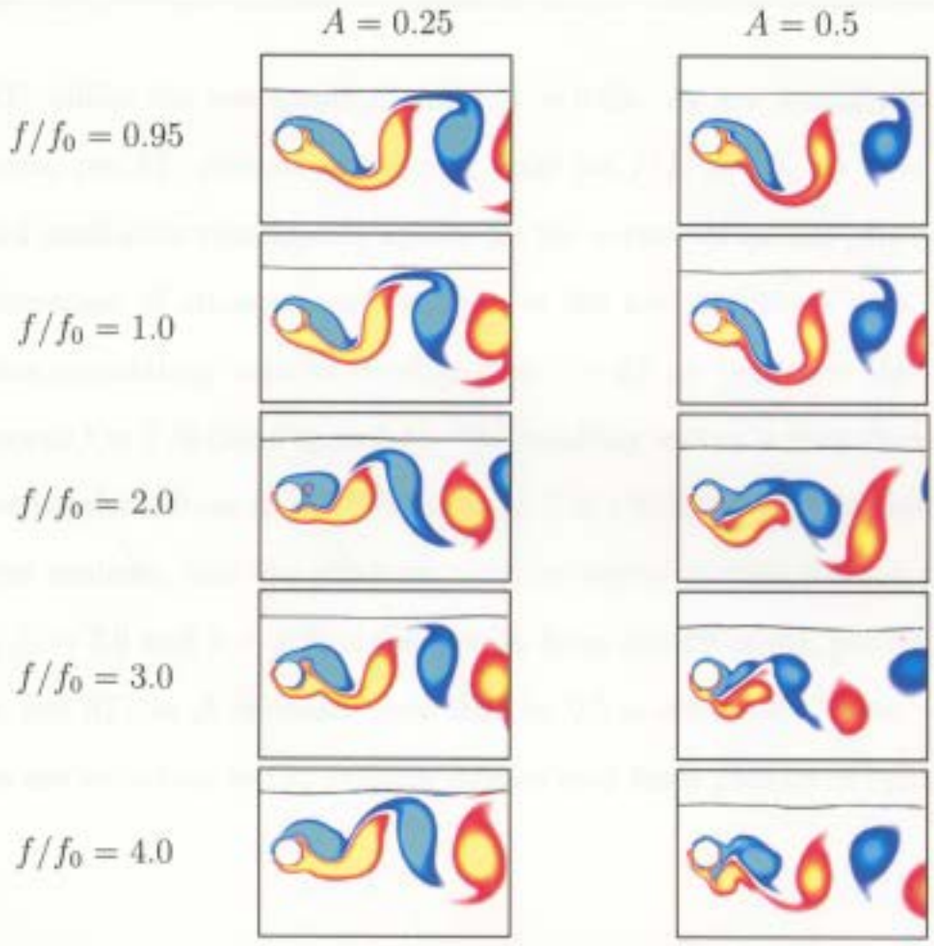

Figure 5.1: The effect of the frequency ratio, $0.95 \leqslant f/f_0 \leqslant 4.0$, on the vortex formation for R = 200, $h = 1.25$: $A = 0.25$ (left) and $A = 0.5$ (right). All equivorticity patterns are taken at instant $y(t) = A$.

and 0.5.

Finally, for selected values of $f/f_0 = 2.0$ and 3.0, the effect of the transverse oscillation amplitude, A, on the equivorticity patterns in the presence of a free surface at $h = 1.25$ is considered. Figures 5.2 and 5.3 show the equivorticity patterns over $5T$ for $f/f_0 = 2.0$ and over $3T$ for $f/f_0 = 3.0$, respectively. At $A = 0.25$ and $f/f_0 = 2.0$ the equivorticity patterns display the asymmetric **2S** mode, per $2T$, of vortex shedding. However, at higher amplitude $A = 0.5$, the immediate coalescence of two co-rotating vortices in the very near wake region is observed. The vortex shedding mode is **C(4S)**,

per $4T$, unlike the low amplitude case $A = 0.25$. At low amplitude, $A = 0.25$, the **2S** mode, per $3T$, persists in the near wake for $f/f_0 = 3.0$. At $A = 0.5$, the strong upward oscillation significantly speeds up the vortex formation process, this leads to the formation of stronger vortices than in the low amplitude case. A pair of two negative co-rotating vortices developed at $t = 0T$ on the lower side of the cylinder coalesce at $t \approx T/2$ (see Figure 5.3). The resulting vortex is then shed at $t \approx 2T$. The inverse process occurs at the time interval $T \leqslant t \leqslant 3T/2$, when a pair of two positive vortices coalesce, and the resulting positive vortex is then shed at $t \approx 2T$. Thus, for $f/f_0 = 3.0$ and $h = 1.25$ a changeover from the **2S** mode, per $3T$, to the **C(2S)** mode, per $3T$, as A increases from 0.25 to 0.5 is observed. These vortex shedding modes are locked-on to the cylinder motion over three periods of cylinder oscillation.

5.2 Future work

It is believed that this thesis research has contributed to the non-boundary fitted Cartesian grid numerical simulation techniques and also shed some light on certain aspects of the free surface flows involving a transversely oscillating cylinder. It is believed that the computational model of this thesis, based on the extension of Reynolds transport theorem for control volumes containing a fluid interface, paves the way for attacking a wide range of free surface flows involving moving rigid bodies, numerically.

This thesis has also raised a number of issues that require further investigation and improvement. It is noted that with the assumptions made in the approximations of

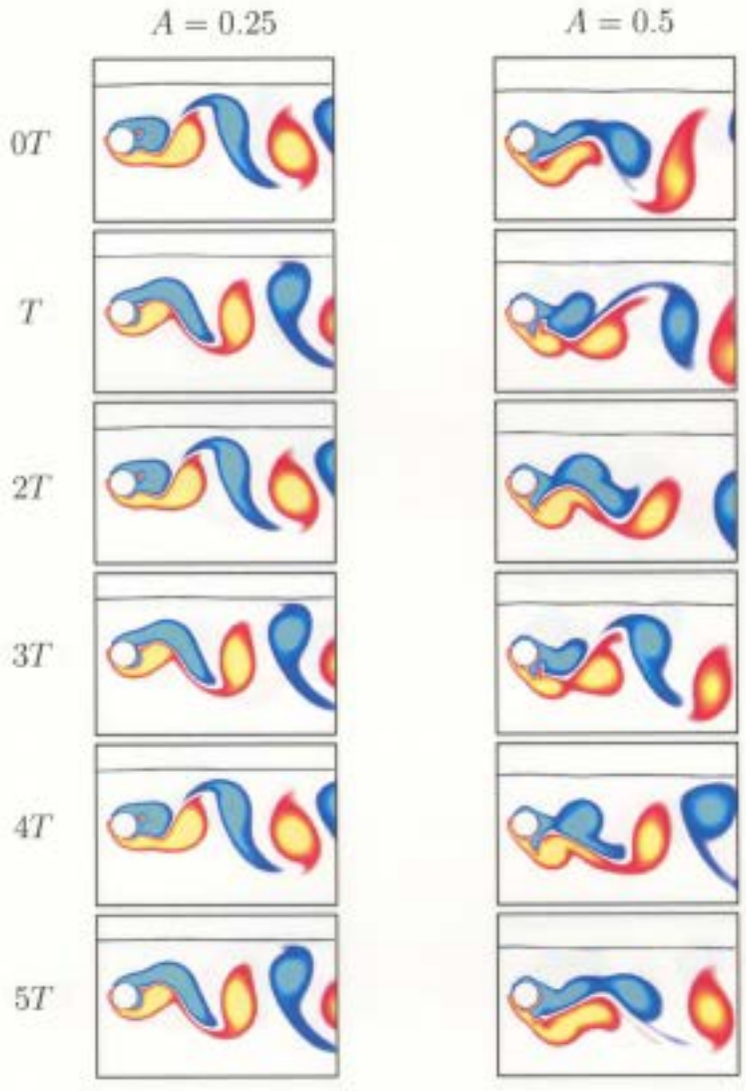

Figure 5.2: The effect of the transverse amplitude, A, on the equivorticity patterns for R = 200, $h = 1.25$ and $f/f_0 = 2.0$ ($T \approx 2.62$). $A = 0.25$: $102.6 \leqslant t \leqslant 115.6$ (left) and $A = 0.5$: $100.6 \leqslant t \leqslant 113.8$ (right). All equivorticity patterns are taken at instant $y(t) = A$.

the free surface velocity boundary conditions discussed in Sections 2.5.2 and 2.5.3, it is not always possible to enforce the exact conservation of mass at the free surface. Under certain conditions, this may even cause a breakdown of computations due to unphysical velocity behaviour at the free surface. This behaviour is also reported

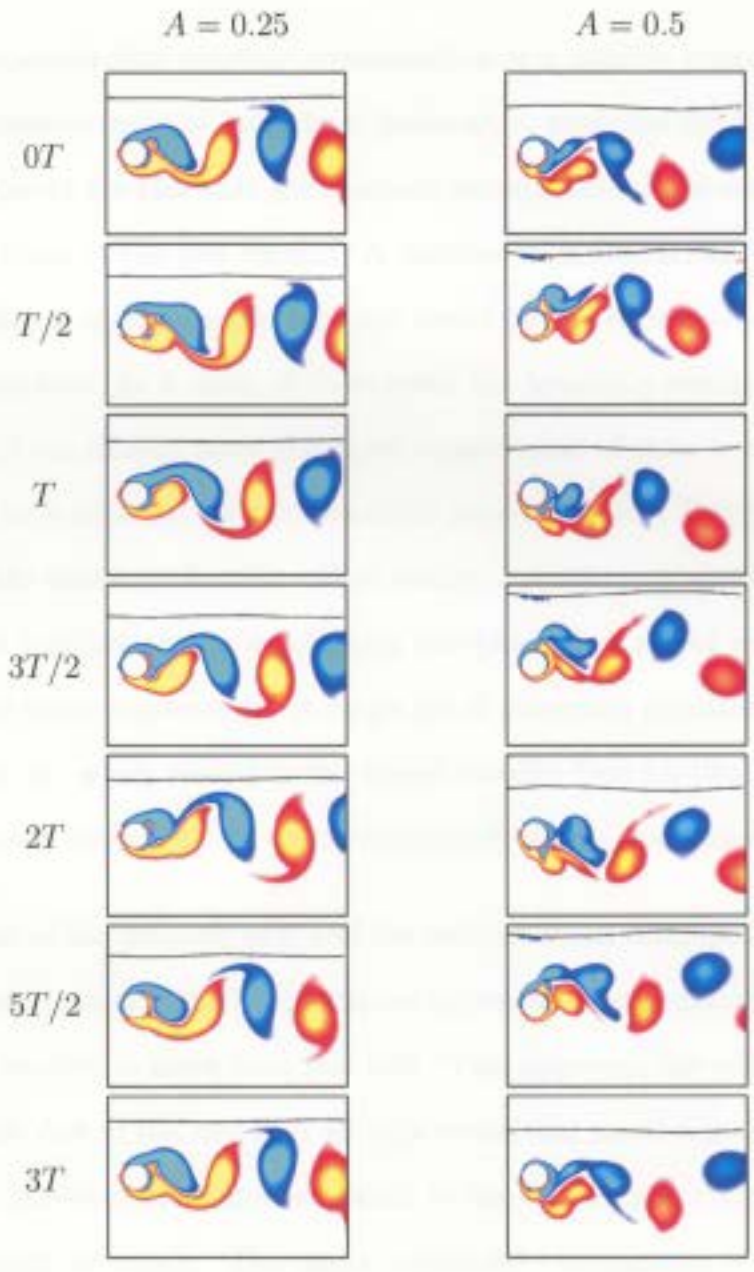

Figure 5.3: The effect of the transverse amplitude, A, on the equivorticity patterns for R = 200, $h = 1.25$ and $f/f_0 = 3.0$ ($T \approx 1.75$). $A = 0.25$: $112.0 \leqslant t \leqslant 117.2$ (left) and $A = 0.5$: $77.66 \leqslant t \leqslant 82.91$ (right). All equivorticity patterns are taken at instant $y(t) = A$.

in the works of Fekken (2004) and Kleefsman (2005), who attempted to overcome this problem by using constant extrapolation of fluid velocities at the free surface.

These studies reported that constant extrapolation is a reliable approach that does not lead to unphysical velocity behaviour; however, it produces unphysical pressure spikes. This is due to the fact that the constant extrapolation does not lead to exact conservation of mass at the free surface. A number of modifications of the velocity boundary conditions at the free surface are tested in the course of developing the present numerical tool. As a result of these tests, the boundary conditions suggested by Gerrits (2001) are chosen, since the exact conservation of mass is enforced whenever possible, which does not lead to numerical pressure spikes. Reichl *et al.* (2005) suggested that the direct application of free surface boundary conditions can be substituted with an implicit one by considering two-phase flow model which takes the motion of the air into consideration. A single set of governing equations is applied to the fluid and the air, which results in the shared velocity field for two fluid layers. In this approach, mass will always be conserved exactly at the free surface.

The classification of the pressure cells and the velocity knots described in Section 2.3 protects the discretization of the Navier-Stokes equations in computational cells where the solid body fraction is more than one half. This improves the robustness of the numerical scheme due to the fact that all cells containing small amounts of fluid are not considered. However, this approach leads to the loss of geometrical information near the fluid-body interface. The major undesired consequence is the numerical error introduced in the computations of the lift and drag forces acting on the body. An improved classification of the pressure cells and the velocity knots is presented in Fekken's work (see Fekken (2004)) along with additional measures to prevent the sharpening of the stability conditions that may arise due to a very small fractional areas and volumes in the case of a body moving through the computational grid.

Even more accurate results can be obtained (see, for example, Udaykumar *et al.* (2001), Kirkpatrick *et al.* (2002)) at the expense of a substantial complication in the discretization due to the introduction of the cell-merging (or cell-linking) techniques.

Bibliography

Alexandrescu, A. (2001). *Modern c++ Design.* Addison-Wesley, New York.

Al-Mdallal, Q. M. (2004). *Analysis and computation of the cross-flow past an oscillating cylinder with two degrees of freedom.* PhD Thesis, Memorial University of Newfoundland, St. John's, Canada.

Anagnostopoulos, P. (2000). Numerical study of the flow past a cylinder excited transversely to the incident stream. Part1: Lock-in zone, hydrodynamic forces and wake geometry. *Journal of Fluids and Structures*, 14:819-851.

Anagnostopoulos, P. (2002). *Flow-Induced Vibrations in Engineering Practice.* WIT Press, UK.

Ananthakrishnan, P. (1991). *Surface waves generated by a translating two-dimensional body: Effects of viscosity.* PhD Thesis, University of California, California.

Anderson, J. D., Jr. (1995). *Computational Fluid Dynamics: The Basics with Applications.* McGraw-Hill, UK.

Aulisa, E., Scardovelli, R., Manservisi, S. and Zaleski, S. (2003). A geometrical area-preserving volume of fluid advection method. *Journal of Computational Physics*, 192(1):355–364.

Bearman, P. (1984). Vortex shedding from oscillating bluff bodies. *Annual Review of Fluid Mechanics*, 16:195-222.

Benzi, M. (2002). Preconditioning techniques for large linear systems: A survey. *Journal of Computational Physics*, 182:418-477.

Berger, M. and Oliger, J. (1984). Adaptive mesh refinement for hyperbolic partial differential equations. *Journal of Computational Physics*, 53:484-512.

Blackburn, H. M. and Henderson, R. D. (1999). A study of two-dimensional flow past an oscillating cylinder. *Journal of Fluid Mechanics*, 385:255-286.

Booch, G. (1994). *Object-Oriented Analysis and Design with Applications* (2nd ed.). Addison-Wesley, New York.

Braza, M., Chassaing, P. and Ha Minh, H. (1986). Numerical study and physical analysis of the pressure and velocity fields in the near wake of a circular cylinder. *Journal of Fluid Mechanics*, 165:79-130.

Brent, R. P. (1973). *Algorithms for Minimization without Derivatives.* Prentice-Hall, New Jersey.

Carberry, J. (2002). *Wake states of an oscillating cylinder and of a cylinder beneath a free-surface.* PhD Thesis, Monash University, Clayton, Australia.

Carberry, J., Sheridan, J. and Rockwell, D. (2001). Stationary and oscillating cylinders in the presence of a free-surface. *Proceedings of the 14th Australasian Fluid Mechanics Conference,* 285-788. Adelaide, Australia.

Cesareo de La Rosa Siqueira. (2006). *Von Kármán vortex street.* Retrieved on 25 November 2006 from http://en.wikipedia.org/wiki/Von_Kármán_vortex_street.

Cetiner, O. and Rockwell, D. (2001). Streamwise oscillations of a cylinder in a steady current. Part 2: Free-surface effects on vortex formation and loading. *Journal of Fluid Mechanics*, 427:29-59.

Choi, S., Choi, H. and Kang, S. (2002). Characteristics of flow over a rotationally oscillating cylinder at low Reynolds number. *Physics of Fluids*, 14(8):2767-2777.

Chung, T. (2002). *Computational Fluid Dynamics.* Cambridge University Press, UK.

Farassat, F. (1996). Introduction to generalized functions with applications in aerodynamics and aeroacoustics, NASA Technical Paper No. 3428. *Langley Research Center.* Hampton, Virginia.

Fekken, G. (2004). *Numerical simulation of free-surface flow with moving rigid bodies.* PhD Thesis, Rijksuniversiteit Groningen, Groningen, The Netherlands.

Flemming, F. and Williamson, C. H. K. (2005). Vortex-induced vibrations of a pivoted cylinder. *Journal of Fluid Mechanics*, 522:215-252.

Fletcher, C. A. (1988). *Computational Techniques for Fluid Dynamics.* Springer-Verlag, New York.

Gerrits, J. (2001). *Dynamics of liquid-filled spacecraft.* PhD Thesis, Rijksuniversiteit Groningen, Groningen, The Netherlands.

Glowinski, R., Pan, T., Hesla, T. I., Joseph, D. D. and Periaux, J. (2001). A fictitious domain approach to the direct numerical simulation of incompressible viscous flow past moving rigid bodies: Application to practical flow. *Journal of Computational Physics*, 169:363-426.

Gresho, P. M. and Sani, R. L. (1998). *Incompressible Flow and the Finite Element Method.* John Wiley and Sons Ltd., New York.

Griffin, O. and Hall, M. (1991). Review-vortex shedding lock-on and flow control in bluff body wakes. *Journal of Fluids Engineering*, 113:526-537.

Harlow, F. H. and Welsh, J. E. (1965). Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *The Physics of Fluids*, 8:2182-2189.

Henderson, R. D. (1997). Nonlinear dynamics and patterns in turbulent wake transitions. *Journal of Fluid Mechanics*, 352:65-112.

Hirt, C. W. (1992). Volume-fraction techniques: Powerful tools for flow modeling, Technical Report No. FSI-92-00-02. *Flow Science Inc.* Santa Fe, New Mexico.

Hirt, C. W. and Nichols, B. D. (1981). Volume of fluid method for the dynamics of free boundaries. *Journal of Computational Physics*, 39:201-225.

Hirt, C. W. and Sicilian, J. M. (1985). A porosity technique for the definition of obstacles in rectangular cell meshes. *Proceedings of the* 4^{th} *International Conference on Ship Hydrodynamics.* Washington, District of Columbia.

King, R. (1977). A review of vortex shedding research and its applications. *Ocean Engineering*, 4:141-172.

Kirkpatrick, M., Armfield, S. and Kent, J. (2002). A representation of curved boundaries for the solution of the Navier-Stokes equations on a staggered three-dimensional Cartesian grid. *Journal of Computational Physics*, 184:1-36.

Kleefsman, T. (2005). *Water impact loading on offshore structures.* PhD Thesis, Rijksuniversiteit Groningen, Groningen, The Netherlands.

Leonard, B. (1979). A stable and accurate convective modelling procedure based on quadratic upstream interpolation. *Computational Methods in Applied Mechanical Engineering*, 19:59-98.

Li, Y., Shock, R., Zhang, R. and Chen, H. (2004). Numerical study of flow past an impulsively started cylinder by the lattice-Boltzmann method. *Journal of Fluid Mechanics*, 519:273-300.

Lomax, H., Pulliam, T. and Zingg, D. (2001). *Fundamentals of Computational Fluid Dynamics.* Springer-Verlag, New York.

Meneghini, J. R. and Bearman, P. W. (1995). Numerical simulation of high amplitude oscillatory flow about a circular cylinder. *Journal of Fluids and Structures*, 9:435-455.

Naudascher, E. and Rockwell, D. (1993). *Flow-Induced Vibrations. An Engineering Guide.* A.A. Balkema Publishers, The Netherlands.

Ongoren, A. and Rockwell, D. (1988). Flow structure from an oscillating cylinder. Part 1. Mechanics of phase shift and recovery in the near wake. *Journal of Fluid Mechanics*, 191:197-223.

Osher, S. and Sethian, J. (1988). Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics.*, 79:12-49.

Patankar, S. V. (1980). *Numerical Heat Transfer and Fluid Flow.* Hemisphere Publishing, Columbia.

Peskin, C. S. (1977). Numerical analysis of blood flow in the heart. *Journal of Computational Physics*, 25:220-252.

Pilliod, J. E., Jr. and Puckett, E. G. (2004). Second-order accurate volume of fluid algorithms for tracking material interfaces. *Journal of Computational Physics*, 199(2):465–502.

Ploumhans, P. and Winckelmans, G. S. (2000). Vortex methods for high-resolution simulations of viscous flow past bluff bodies of general geometry. *Journal of Computational Physics*, 165:354-406.

Poncet, P. (2004). Topological aspects of three-dimensional wakes behind rotary oscillating cylinders. *Journal of Fluid Mechanics*, 517:27-53.

Reichl, P. J. (2001). *Flow past a cylinder close to a free surface.* PhD Thesis, Monash University, Clayton, Australia.

Reichl, P. J., Hourigan, K. and Thompson, M. (2005). Flow past a cylinder close to a free surface. *Journal of Fluid Mechanics*, 533:269-296.

Rider, W. J. and Kothe, D. B. (1998). Reconstructing volume tracking. *Journal of Computational Physics*, 141(2):112–152.

Riley, N. and Yan, B. (1996). Boundary-layer around a submerged circular cylinder induced by free-surface travelling waves. *Journal of Fluid Mechanics*, 316:241-257.

Rockwell, D. (1998). Vortex-body interactions. *Annual Review of Fluid Mechanics*, 30:199-229.

Rood, E. P. (1995). Vorticity interractions with a free surface. In: *Fluid Vortices* (edited by S. I. Green), 687-730. Kluwer Academic Publishers, UK.

Sarpkaya, T. (2004). A critical review of the intrinsic nature of vortex induced vibrations. *Journal of Fluids and Structures*, 19(4):389-447.

Scardovelli, R. and Zaleski, S. (1999). Direct numerical simulation of free surface and interfacial flow. *Annual Review of Fluid Mechanics*, 31:567-603.

Serrin, J. (1959). Mathematical principles of classical fluid dynamics. In: *Encyclopedia of Physics.* Springer-Verlag, New York.

Sethian, J. A. and Smereka, P. (2003). Level set methods for fluid interfaces. *Annual Review of Fluid Mechanics*, 35:341-372.

Sheridan, J., Lin, J. C. and Rockwell, D. (1997). Flow past a cylinder close to a free surface. *Journal of Fluid Mechanics*, 330:1-30.

Stroustrup, B. (2000). *The C++ Programming Language (Special Edition).* Addison-Wesley, New York.

Stroustrup, B. (2004). Abstraction and the C++ machine model. *Proceedings of the International Conference on Embedded Software and Systems,* 1-13. Hangzhou, China.

Sumer, B. M. and Fredsøe, J. (1997). *Hydrodynamics Around Cylindrical Structures.* World Scientific, UK.

Udaykumar, H., Mittal, R., Rampunggoon, P. and Khanna, A. (2001). A sharp interface Cartesian grid method for simulating flows with complex moving boundaries. *Journal of Computational Physics*, 174:345-380.

Vandevoorde, D. and Josuttis, N. M. (2002). *C++ Templates.* Addison-Wesley, New York.

Verstappen, R. W. C. P. and Veldman, A. E. P. (2000). Numerical computation of viscous flow around a circular cylinder on a Cartesian grid. *Proceedings of the European Congress on Computational Methods in Applied Sciences and Engineering.* Barcelona, Spain.

Versteeg, H. K. and Malalasekera, W. (1995). *An Introduction to Computational Fluid Dynamics. The Finite Volume Method.* Longman Scientific and Technical, UK.

Williamson, C. (1996). Vortex dynamics in the cylinder wake. *Annual Review of Fluid Mechanics*, 28:477-539.

Williamson, C. and Govardhan, R. (2004). Vortex-induced vibrations. *Annual*

Review of Fluid Mechanics, 36:413-455.

Williamson, C. H. K. and Roshko, A. (1988). Vortex formation in the wake of an oscillating cylinder. *Journal of Fluids and Structures*, 2:355-381.

Yan, B. (2000). Unsteady viscous flow about a submerged circular cylinder with a steady current. *Fluid Dynamics Research*, 26:69-94.

Zdravkovich, M. (1997). *Flow Around Circular Cylinders: Vol. 1: Fundamentals.* Oxford University Press, UK.

Zhu, Q., Lin, J. C., Unal, F. M. and Rockwell, D. (2000). Motion of a cylinder adjacent to a free surface: Flow patterns and loading. *Experiments in Fluids*, 28:559-575.

Appendix A

Navier-Stokes equations in a non-inertial frame of reference

Consider the following form of the Navier-Stokes equations

$$\frac{\partial u_j}{\partial x_j} = 0, \tag{A.1a}$$

$$\frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{1}{\mathrm{R}} \frac{\partial^2 u_i}{\partial x_j \partial x_j} + g_j \tag{A.1b}$$

which are valid in any inertial frame of reference (j is a dummy summation index). Here p denotes scaled pressure $p' \equiv p/\rho$.

Let $\mathbb{X}$ be an inertial frame of reference with associated coordinates $(\underline{x}, t)$. Consider another frame of reference $\hat{\mathbb{X}}$ with corresponding coordinates $\hat{\underline{x}}, \hat{t}$ which is allowed to move without rotation with respect to $\mathbb{X}$. In order to derive the Navier-Stokes equa-

tions in terms of the variables $(\underline{\hat{x}}, \hat{t})$, all variables and partial derivatives corresponding to $\mathbb{X}$ are to be expressed in terms of variables and partial derivatives corresponding to $\hat{\mathbb{X}}$ and then substituted into equations (A.1).

We start by stating relations between the coordinates. Relativity effects are not considered, so the time variables are assumed to be identical in both frames of reference

$$\hat{t}(\underline{x}, t) = t. \tag{A.2}$$

Let $\underline{V}(t)$ be the velocity at the origin of $\hat{\mathbb{X}}$ relative to the inertial frame. Since $\hat{\mathbb{X}}$ moves without rotation with respect to $\mathbb{X}$, the space coordinates and velocities are related as follows

$$\hat{x}_i(\underline{x}, t) = x_i - \int_0^t V_i(\tau)\, \mathrm{d}\tau\,, \tag{A.3}$$

$$\hat{u}_i = u_i - V_i\,. \tag{A.4}$$

Let function $F(\underline{\hat{x}}, \hat{t})$ be a fluid property. We wish to calculate partial derivatives of this function in terms of $\underline{x}$ and t. We note that $\underline{\hat{x}}$ and $\hat{t}$ depend on $\underline{x}$ and t: $F = F(\underline{\hat{x}}(\underline{x}, t), \hat{t}(\underline{x}, t))$. Then, the partial derivatives of F with respect to time and space can be computed according to the theorem of differentiation of a composite function. Consider a point which has coordinates $(\underline{x_0}, t_0)$ in $\mathbb{X}$ and coordinates $(\underline{\hat{x}}_0, \hat{t}_0)$ in $\hat{\mathbb{X}}$. Then, according to the theorem,

$$\frac{\partial F}{\partial x_i}(\underline{x}_0, t_0) = \frac{\partial F}{\partial \hat{x}_j}(\underline{\hat{x}}_0, \hat{t}_0)\, \frac{\partial \hat{x}_j}{\partial x_i}(\underline{x}_0, t) + \frac{\partial F}{\partial \hat{t}}(\underline{\hat{x}}_0, \hat{t}_0) \frac{\partial \hat{t}}{\partial x_i}(\underline{x}_0, t), \tag{A.5}$$

$$\frac{\partial F}{\partial t}(\underline{x}_0, t_0) = \frac{\partial F}{\partial \hat{x}_j}(\underline{\hat{x}}_0, \hat{t}_0)\ \frac{\partial \hat{x}_j}{\partial t}(\underline{x}_0, t) + \frac{\partial F}{\partial \hat{t}}(\underline{\hat{x}}_0, \hat{t}_0)\frac{\partial \hat{t}}{\partial t}(\underline{x}_0, t)\,. \tag{A.6}$$

We can then use equations (A.2) and (A.3) to compute the following partial derivatives

$$\begin{aligned} \frac{\partial \hat{t}}{\partial t} &= \frac{\partial t}{\partial t} = 1\,, & \frac{\partial \hat{t}}{\partial x_i} &= \frac{\partial t}{\partial x_i} = 0\,; \\ \frac{\partial \hat{x}_i}{\partial x_j} &= \delta_{i,j} - 0 = \delta_{i,j}\,, & \frac{\partial \hat{x}_i}{\partial t} &= -V_i\,. \end{aligned} \tag{A.7}$$

Substitution of (A.7) into (A.5) and (A.6) results in

$$\frac{\partial F}{\partial x_i}(\underline{x}_0, t_0) = \frac{\partial F}{\partial \hat{x}_i}(\underline{\hat{x}}_0, \hat{t}_0)\,, \tag{A.8}$$

$$\frac{\partial F}{\partial t}(\underline{x}_0, t_0) = -\frac{\partial F}{\partial \hat{x}_j}(\underline{\hat{x}}_0, \hat{t}_0)\ V_j + \frac{\partial F}{\partial \hat{t}}(\underline{\hat{x}}_0, \hat{t}_0)\,. \tag{A.9}$$

Equations (A.4) and (A.8) with $F = u$ are used to derive

$$\frac{\partial u_j}{\partial x_j} = \frac{\partial \hat{u}_j}{\partial \hat{x}_j} + \frac{\partial V_j}{\partial \hat{x}_j} = \frac{\partial \hat{u}_j}{\partial \hat{x}_j} + 0 = \frac{\partial \hat{u}_j}{\partial \hat{x}_j}\,, \tag{A.10}$$

which is then substituted into (A.1a) to obtain the continuity equation in the non-inertial frame of reference

$$\frac{\partial \hat{u}_j}{\partial \hat{x}_j} = 0\,. \tag{A.11}$$

Now, consider equation (A.1b). By using (A.4) and (A.9), the first (unsteady) term is reduced to

$$\frac{\partial u_i}{\partial t} = -\frac{\partial(\hat{u}_i + V_i)}{\partial \hat{x}_j}V_j + \frac{\partial \hat{u}_i}{\partial \hat{t}} + \frac{\partial V_i}{\partial \hat{t}} = -\frac{\partial \hat{u}_i}{\partial \hat{x}_j}V_j + \frac{\partial \hat{u}_i}{\partial \hat{t}} + \frac{\partial V_i}{\partial \hat{t}}\,. \tag{A.12}$$

Due to (A.4) and (A.8), the second (convection) term is reduced to

$$u_j \frac{\partial u_i}{\partial x_j} = \hat{u}_j \frac{\partial \hat{u}_i}{\partial \hat{x}_j} + V_j \frac{\partial \hat{u}_i}{\partial \hat{x}_j} \,. \tag{A.13}$$

Similarly, (A.4) and (A.8) are used to transform the right hand side of equation (A.1b). Terms involving V_j in equations (A.12) and (A.13) cancel each other, and the momentum equation in the non-inertial frame of reference takes the following form

$$\frac{\partial \hat{u}_i}{\partial \hat{t}} + \hat{u}_j \frac{\partial \hat{u}_i}{\partial \hat{x}_j} + \frac{\partial V_i}{\partial \hat{t}} = -\frac{\partial p}{\partial \hat{x}_i} + \frac{1}{\mathrm{R}} \frac{\partial^2 \hat{u}_i}{\partial \hat{x}_j \partial \hat{x}_j} + g_j \,. \tag{A.14}$$

Equation (A.14) has one extra term $\left(\frac{\partial V_i}{\partial \hat{t}}\right)$ when compared with (A.1b). Since this term is the acceleration of the non-inertial frame of reference with respect to the inertial one, it is usually included in the volumetric force term

$$\frac{\partial \hat{u}_i}{\partial \hat{t}} + \hat{u}_j \frac{\partial \hat{u}_i}{\partial \hat{x}_j} = -\frac{\partial p}{\partial \hat{x}_i} + \frac{1}{\mathrm{R}} \frac{\partial^2 \hat{u}_i}{\partial \hat{x}_j \partial \hat{x}_j} + \left(g_j - \frac{\partial V_i}{\partial \hat{t}}\right) \,. \tag{A.15}$$

Thus, we have proved that the Navier-Stokes equations do not change their form in a non-inertial frame of reference except for the volumetric force term, which is modified to account for the acceleration of the non-inertial frame of reference.

It is noted that this result is valid for arbitrary motion of a non-inertial frame of reference. This is illustrated using a physical approach. Acceleration of a fluid particle in an inertial frame is a sum of the acceleration of that particle in a non-inertial frame and the acceleration of the non-inertial frame relative to the inertial one

$$\underline{a} = \hat{\underline{a}} + \underline{a}_{\mathrm{rel}} \,. \tag{A.16}$$

Fluid particle acceleration is the total time derivative of the fluid velocity

$$\underline{\hat{a}} = \frac{d\underline{\hat{u}}}{dt}. \tag{A.17}$$

Newton's second law is valid in an inertial frame of reference

$$\underline{a} = \Sigma \underline{f}\,, \tag{A.18}$$

where the right hand side is a sum of the surface and volumetric forces acting on the material volume divided by its mass. Substitution of (A.16) into (A.18) results in

$$\frac{d\underline{\hat{u}}}{dt} = \Sigma \underline{f} - \underline{\hat{a}}\,. \tag{A.19}$$

Equation (A.19) is another form of equation (A.15).